Dissertation im Fachbereich Mathematik der Technischen Universität Kaiserslautern

Grey-Box Modelling for Nonlinear Systems

Jan Hauth

Dezember 2008

1. Gutachter: Prof. Dr. Dieter Prätzel-Wolters, Technische Universität Kaiserslautern
2. Gutachter: Prof. Dr. Jürgen Franke, Technische Universität Kaiserslautern

Datum der Disputation: 5. Juni 2008

Vom
Fachbereich Mathematik der Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.)
genehmigte Dissertation

D 386

To my parents Irma born Scholtes and Kurt Hauth

In memoriam Prof. Walter Blankenheim

Abstract

Grey-box modelling deals with models which are able to integrate the following two kinds of information: qualitative (expert) knowledge and quantitative (data) knowledge, with equal importance. The doctoral thesis has two aims: the improvement of an existing neuro-fuzzy approach (LOLIMOT algorithm), and the development of a new model class with corresponding identification algorithm, based on multiresolution analysis (wavelets) and statistical methods. The identification algorithm is able to identify both hidden differential dynamics and hysteretic components.

After the presentation of some improvements of the LOLIMOT algorithm based on readily normalized weight functions derived from decision trees, we investigate several mathematical theories, i.e. the theory of nonlinear dynamical systems and hysteresis, statistical decision theory, and approximation theory, in view of their applicability for grey-box modelling. These theories show us directly the way onto a new model class and its identification algorithm. The new model class will be derived from the local model networks through the following modifications: Inclusion of non-Gaussian noise sources; allowance of internal nonlinear differential dynamics represented by multi-dimensional real functions; introduction of internal hysteresis models through two-dimensional "primitive functions"; replacement respectively approximation of the weight functions and of the mentioned multi-dimensional functions by wavelets; usage of the sparseness of the matrix of the wavelet coefficients; and identification of the wavelet coefficients with Sequential Monte Carlo methods. We also apply this modelling scheme to the identification of a shock absorber.

Abstrakt

Grey-Box-Modellierung beschäftigt sich mit Modellen, die in der Lage sind folgende zwei Arten von Information über ein reales System gleichbedeutend einzubeziehen: qualitatives (Experten-)Wissen, und quantitatives (Daten-)Wissen. Die Dissertation hat zwei Ziele: die Verbesserung eines existierenden Neuro-Fuzzy-Ansatzes (LOLIMOT-Algorithmus); und die Entwicklung einer neuen Modellklasse mit zugehörigem Identifikations-Algorithmus, basierend auf Multiskalenanalyse (Wavelets) und statistischen Methoden. Der resultierende Identifikationsalgorithmus ist in der Lage, sowohl verborgene Differentialdynamik als auch hysteretische Komponenten zu identifizieren.

Nach der Vorstellung einiger Verbesserungen des LOLIMOT-Algorithmus basierend auf von vorneherein normalisierten Gewichtsfunktionen, die auf einer Konstruktion mit Entscheidungsbäumen beruhen, untersuchen wir einige mathematische Theorien, das sind die Theorie nichtlinearer Systeme und Hysterese, statistische Entscheidungstheorie and Approximationstheorie, im Hinblick auf deren Anwendbarkeit für Grey-Box-Modellierung. Diese Theorien führen dann auf direktem Wege zu einer neuen Modellklasse und deren Identifikationsalgorithmus. Die neue Modellklasse wird von Lokalmodellnetzwerken durch folgende Modifikationen abgeleitet: Einbeziehung von nicht-Gaußschen Rauschquellen; Zulassung von interner nichtlinearer Differentialdynamik repräsentiert durch mehrdimensionale reelle Funktionen; Einführung interner Hysterese-Modelle mittels zweidimensionaler „Stammfunktionen“; Ersetzung bzw. Approximation der Gewichtsfunktionen und der erwähnten mehrdimensionalen Funktionen durch Wavelet-Koeffizienten; Ausnutzung der Dünnbesetztheit der Wavelet-Koeffizienten-Matrix; und Identifikation der Wavelet-Koeffizienten mit Sequentiellen Monte Carlo-Methoden. Wir wenden dieses Modellierungsschema dann auf die Identifikation eines Stoßdämpfers an.

Contents

Thanks xv

Overview xvii

Notations xxv

1 Introduction: Grey-box models and the LOLIMOT algorithm 1
- 1.1 Systems and models . . . 2
  - 1.1.1 Nonlinear dynamical systems and model schemes . . . 2
  - 1.1.2 Properties of systems and models . . . 6
  - 1.1.3 Separation of dynamics . . . 8
  - 1.1.4 Linear combinations of basis functions and networks . . . 11
- 1.2 Local model networks . . . 14
- 1.3 The LOLIMOT algorithm . . . 19
- 1.4 Problems and possible improvements . . . 25
  - 1.4.1 Decision trees und weight functions . . . 30
  - 1.4.2 Gradient based optimization . . . 44
  - 1.4.3 Applications of the gradient based optimization to the improvement of the LOLIMOT algorithm . . . 57

2 Dealing with time: Dynamics 63
- 2.1 Deterministic models for dynamical systems . . . 64
- 2.2 Preisach hysteresis . . . 74
  - 2.2.1 Definition and properties . . . 75
  - 2.2.2 Implementation . . . 93
  - 2.2.3 Identification . . . 109
- 2.3 Conclusions . . . 113

3 Stochastic decision theory: Bridge between theory and reality 117
- 3.1 Models for reality . . . 118
- 3.2 Bayesian statistics . . . 123
  - 3.2.1 Bayes' theorem . . . 124
  - 3.2.2 Foundations of decision theory . . . 126
  - 3.2.3 Justifications for Bayesian inference . . . 135
- 3.3 Priors . . . 140
  - 3.3.1 Strategies for prior determination . . . 140
  - 3.3.2 Hierarchical Bayes . . . 147

3.4 Stochastic models and Bayesian estimation . . . 150
3.4.1 Static normal models . . . 150
3.4.2 Dynamic models . . . 153
3.4.3 Markov chains . . . 158
3.4.4 Graphical models . . . 168
3.5 Computational issues . . . 170
3.5.1 Bayesian calculations . . . 170
3.6 State space systems and recursive computations . . . 183
3.6.1 General state space models . . . 183
3.6.2 Filtering and smoothing . . . 186
3.6.3 Exact algorithms for filtering and smoothing . . . 192
3.6.4 Approximations . . . 194

4 Signal processing, representation and approximation: Wavelets 209
4.1 Wavelets . . . 211
4.1.1 Signal analysis . . . 211
4.1.2 Time-scale wavelets . . . 215
4.1.3 The continuous wavelet transform . . . 217
4.1.4 The discrete wavelet transform . . . 219
4.1.5 Multiresolution analysis and Fast Wavelet Transform (FWT) . . . 221
4.1.6 Wavelet packets . . . 231
4.2 Nonlinear approximation . . . 234
4.2.1 Approximation theory . . . 236
4.2.2 Approximation and wavelets . . . 248
4.2.3 Highly nonlinear approximation . . . 254
4.3 Wavelets and Bayesian techniques: Denoising . . . 262
4.4 Wavelets and dynamical systems . . . 275
4.4.1 Nonparametric estimation . . . 275
4.4.2 Linear systems and frames . . . 275

5 Putting things together: Implementation and application 277
5.1 Summary . . . 278
5.2 Model, algorithm and implementation . . . 281
5.2.1 Model . . . 282
5.2.2 Algorithm . . . 285
5.2.3 Implementation . . . 292
5.3 Examples . . . 292
5.3.1 First example: Linear mass-spring-damper system . . . 293
5.3.2 Second example: Nonlinear mass-spring-damper system . . . 298
5.3.3 Third example: Preisach hysteresis . . . 300
5.4 Identification of real data . . . 303
5.4.1 The data . . . 303
5.5 Conclusion and future work . . . 312
5.5.1 Résumé: Usage of identification methods . . . 312

5.5.2 Future work 317
5.5.3 Conclusion 317

Appendix: Basic notions **319**

Bibliography **327**

Index of definitions **347**

Schritt — Atemzug — Besenstrich

Beppo Straßenkehrer in Michael Ende's *Momo*

Thanks

My thanks go to the Fraunhofer Institut für Techno- und Wirtschaftsmathematik (ITWM; Institute for Industrial Mathematics) in Kaiserslautern, Germany, for the provision of room and means, and in particular to the head of this institute Prof. Dr. Dieter Prätzel-Wolters, my Doktorvater, and to Dr. Patrick Lang, head of the department of Adaptive Systems of the same Institute, both for supervising me during my doctoral transition steps. My thank goes also to the Graduiertenkolleg Mathematik und Praxis (Graduate Research Training Programme Mathematics and Practice) which provided a scholarship to me during the years 2003-2006, and especially to its head Prof. Dr. Jürgen Franke for his valuable hints. I also want to give thanks to my industry partner, the company LMS Deutschland GmbH, Kaiserslautern, and especially to Dr. Manfred Bäcker, who always was very interested in the results of this thesis and always willing in providing measurement data to test the models.

Thanks go also to the department MDF of the ITWM, especially to Michael Speckert who prepared simulation data to test the models (which I regrettably could not manage to include into my thesis before finishing it).

Thanks go also to my former co-doctorands Eva Barrena Algara (who in discussions about mathematics and other things was persistent enough to finally persuade me of my being wrong), to Dr. Beatriz Rasero Clavero, to Dr. Frank Kneip, and to Dr. Bernd Büchler for valuable discussions, hints, ideas, and their friendship.

My most affectionate thanks go to my parents to whom this thesis is dedicated, just for their being parents one can only dream of, to my sister Tina, to whom an analogue statement holds, and to my dearest love Anne Fuchs, whom I thank not only for her funny Waliationen, but for many invaluable moments in my life. I apologize for all their sufferings I caused them during the last years when they only stood at the second place after my thesis.

In March 2007, Prof. Walter Blankenheim died ten days before the beginning of the 6th edition of the piano competition “J.S.Bach” Saarbrücken, now Würzburg, which he was the founder and head of. It was always a great pleasure for me to help him organizing these competitions and to attend his Bach piano courses as a passive participator. I am glad that his colleague Prof. Inge Rosar will continue the competitions, and I wish her all the best for this!

This thesis was partly funded by the “Stiftung Rheinland-Pfalz für Innovation” through the project “Nichtlinear dynamische Bauteilmodelle für die Fahrzeugsimulation”.

Overview

Grey-box models for nonlinear systems

The task of building mathematical models is the translation of certain interesting properties of a real system into mathematical equations. To execute this task, it is necessary to have access to information about the real system one wants to model. This information is of three kinds (see e.g. Bernardo [2003]). First of all, it is necessary to have knowledge (K) about the real system, be it structural knowledge on the construction of the system, physical knowledge gained by first principles, or be it historically adopted empirical knowledge. The second kind of information is data (D) taken from deliberately planned experiments on the system. The third kind is more comprehensive: it consists in the assumptions (A) we have to make on the relationship of model and system, for example about the correctness of our model.

Depending on the influences of these three kinds of information during the building of our model, we distinguish roughly between several model types. Since the assumptions (A) of the correctness of our model generally have to be made for all model types, the distinction is along the other two kinds of information, namely knowledge (K) and data (D). Looking at the extreme poles, we on one side have models which are built only by using knowledge (K), called the white-box models, and on the other side, the models which are to be estimated by experimental data (D). In reality, models do not belong to exactly one type. Generally all kinds of information have to be applied, but often there is a tendency towards one of these extrema.

White-box models thus subsume all kinds of (ordinary or partial) differential equations derived from first principles. Another example is the rule-based models like e.g. realized by fuzzy-systems, or more general by expert systems. Conversely, black-box models are mainly given by parameterized model classes, e.g. consist of combinations of simple basis functions respectively basis models (compare Sjöberg et al. [1995] and Juditsky et al. [1995]). The parameters occurring in theses models have to be identified (estimated) through useful data. Into these classes also the neural networks may be counted, which some years ago gained much attention similar to the fuzzy-systems. A problem concerning neural networks as well as fuzzy-systems is that from a mathematical viewpoint, there do not exist well founded underlying theories for their convergence behaviour especially in high dimensions and in the presence of noise. Because of their conceptual simplicity they are nevertheless very popular within the engineering and technical communities.

A third kind of model type is called grey-box model. This type actually subsumes all models in-between the two extremes of white- and black-box models. As already mentioned, a model development process is always driven by both prior knowledge and experimental data, hence principally all models are in some respect grey-box models. As an example we mention models consisting of differential equations where several parameters occurring in these equations are not known and have to be extracted from the system by data-based methods.

Conversely, when models are built by a basis-function approach, one can and has to choose these basis functions according to the properties of the real system. To do the right choice, one needs knowledge about the system.

Grey-box modelling, as we want to have it understood, is able to integrate both kinds of information: qualitative knowledge and quantitative (data) knowledge with equal importance. We have to provide two means: a model class and a corresponding identification algorithm. Both are equally important and have to fit to each other.

In this thesis we want to pursue a mainly non-parametric approach. Non-parametric in the case of model identification means that we do not just estimate a fixed number of scalar parameters. Instead, the models come from classes with a variable number of parameters, and this number (corresponding to the model size) has to be estimated as well. It appears that the methods used in the literature for non-parametric grey-box modelling are not easily treated with mathematical investigations. With the existing methods, every system identification reduces to the solution of the following problem:

- Approximate a multi-dimensional real function which is known only on irregular distributed and finitely many values, these values being additionally disturbed by noise and errors.

These errors are measurement errors which are unavoidable during the data acquisition process (data errors), as well as unmodelled system influences (model errors), or disturbances (from inside or outside). In all cases, it seems that the most natural procedure to handle these in detail unknown errors is the usage of stochastic processes for modelling. Often, errors are simply assumed to be Gaussian noise (if not neglected at all), what in many cases may not be sufficient.

As an example for grey-box modelling we mention the so-called neuro-fuzzy systems. These consist of a combination of fuzzy-system and neural network, and can thus be used for the purpose of grey-box modelling. Both fuzzy-systems and neural networks are used as approximators. In both cases, one tries to approximate a real function which is defined on several variables (multi-dimensional function) by a weighted sum of one-dimensional (and thus easier to handle) real functions. This problem is closely connected with (and is actually a generalization of) Hilbert's 13th problem where a solution (for continuous functions) was given by Kolmogorov and Arnol'd in 1957 (see appendix). This originally purely theoretical concept showed not to be practically applicable, because the mentioned one-dimensional functions could not be computed explicitly. Both fuzzy-systems and neural networks try to overcome these problems.

In the case of fuzzy-systems, predefined data points — these are given through the modelled rule base — are interpolated by fuzzy-logic. Fuzzy-logic is a generalization of the usual two-valued logic through accounting also for intermediate logical values. In contrast, the neural networks can be seen as a formalization of the structure of animal brains. This formalization was given by McCulloch and Pitts already in 1943. Hecht-Nielsen and Funahashi could show in the 1980s that an approximation of a multi-dimensional real function using the so-called three-layered feedforward network, is principally possible. Three-layered neural networks resemble the solution found by Kolmogorov and Arnol'd. There arise several problems: How to find the right complexity of the network (number of neurons)? How to identify the parameters

(weights)? How can errors and disturbances be treated? Only after the introduction of the backpropagation algorithm some success could be achieved. The backpropagation algorithm is nothing else than a local optimization algorithm based on the gradient of the real function realizing the neural network, regarded as function on the parameters. The optimization is thus only done locally and depends strongly on the choice of the initial parameters. The initial parameters are usually chosen randomly, whereas there seldom exist precise descriptions on how to choose them (e.g. with respect to the right probability distributions; without the choice of a probability distribution it is not possible in reality to apply random selections). The original steepest descent method is known to be extremely slow. There are methods which fasten the convergence, like Newton or quasi-Newton methods (see e.g. Sjöberg et al. [1994]), but nevertheless they still seem to be seldom used in applications.

Neuro-fuzzy systems inherit this backpropagation algorithm, but the initial values of the parameters are determined in a more constructive way. Since the parameters are mainly part of the fuzzy-rule system, they can be initialized via a-priori knowledge. Another possibility is the successive addition of new rules, for example by splitting an old rule into two or more derived rules. Also the inverse procedure is reasonable, the merging of two or more rules into one. These split-and-merge algorithms appear also in different settings.

We want to show in this thesis that the mentioned methods (fuzzy-systems coupled with neural networks) can be replaced by other mathematically more grounded ones. Furthermore, after doing this, an essentially larger class of nonlinear systems can be identified. This doctoral thesis has two aims:

- the improvement of an existing neuro-fuzzy approach (LOLIMOT algorithm), and
- the development of a new model class with corresponding identification algorithm, based on multiresolution analysis (wavelets) and statistical methods.

Altogether the following theories play a role:

- theory of nonlinear dynamical systems and hysteresis,
- statistical decision theory,
- approximation theory.

The structure of this thesis is as follows: After the presentation of the improvements of the LOLIMOT algorithm, we want to describe the mentioned theories in the view of their application for grey-box modelling in the subsequent chapters. These theories show us directly the way onto the new model class and its identification algorithm, presented in the last chapter.

Local model networks (chapter 1)

We first want to investigate in this thesis the model class provided by the local model networks which serve equally well for black-box and grey-box models; from the local model networks also the new model class will be derived. The corresponding identification algorithm is the so-called LOLIMOT algorithm (Nelles [2001]). LOLIMOT stands for “LOcal LInear MOdel

Tree". Local model networks (see also Johansen and Murray-Smith [1997]) represent a generalization of the basis-function approach. The basis elements in this case are not functions but (linear) dynamic models. These models are weighted and superposed in such a way that in certain parts of the regime space (this is where the global model lives) mainly only one basis model is "active". In this way the name "local" models for the basis models is justified. The LOLIMOT algorithm produces successively a more and more complex global model: beginning with only one basis model, it iteratively divides the regime space and identifies the newly obtained local models. Since the superposition of the (linear) local models is done through nonlinear weight functions, the overall global model is nonlinear. Principally, the linearity of the basis models — meant is the linearity with respect to the inputs — is not essential for the identification procedure; it is only necessary that the parameters occurring in the basis models are linear. This is also the case for polynomial models. The algorithm works equally well for this kind of models.

We will provide some improvements on the original algorithm. For this reason the weight functions, originally normalized Gaussian bell functions, are replaced by decision-tree based weight functions. These are already normalized, i.e. they form a non-negative partition of the unity. They do not need to be forced to normality. Exactly this procedure leads to some problems when using the Gaussian bells. The introduction of the decision-tree based weight functions enables further improvements: more flexible partitions of the regime space (originally, the only divisions possible have been axis-parallel ones), the application of pruning methods (this means the "resection" of the model tree which originally could only grow) and the introduction of gradient-based optimization procedures. Then the transfer from an overall NARX model to an overall NOE model (better suited for simulation) is possible without problems. The local model networks thus approach even more the neural networks, and similar to their case mathematical convergence results are not available. Nevertheless, an optimal approximation cannot be expected, the algorithm works only suboptimal. Although it seems to be quite stable, at least if linear basis models are used, the restriction to these models may lead to many divisions of the regime space and thus to very complex models. Conversely, the usage of polynomials of higher degree may lead to instabilities during the estimation of the coefficients.

A further extension of the algorithm for the estimation of hysteresis models will be presented in the following chapter.

Theory of nonlinear dynamical systems and hysteresis (chapter 2)

The focus here lies on the nonlinearity. Recently, more and more applications gain attention wherein hysteresis plays a major role (see e.g. Visintin [1994]). Hysteresis in its broadest sense can be seen as the dependence of two signals occurring in the system such that this dependence concerns the whole history of the system. This very broad definition should be sharpened to extreme cases to make investigations possible; otherwise virtually every system could be subsumed under this definition. Therefore one focusses in most cases on rate-independent hysteresis. It is in some way an extreme opposite of the usual systems with differential dynamics: signals which follow rate-independent hysteresis are in a certain way independent of the velocities these signals occur with. Since changes of the velocity are mathematically equally well

described by time transformations, rate-independence is equivalent to the invariance of the system against every time transformation. In reality this extreme case will occur quite seldom, but for theoretical investigations the restriction to this kind of hysteresis enlightens matters. In the literature, the notion of hysteresis is often used in this restricted sense of rate-independent hysteresis. One of the most popular models for (a special kind) of hysteresis is the Preisach model. For an incorporation of Preisach hysteresis into neural network models see Kirchmair [2002].

A further extension of the LOLIMOT algorithm was done by applying it to hysteresis models of Preisach type. Therefore some kind of primitive function of the Preisach function was used, which can be identified with exactly the same methods as the linear models in the original algorithm. Though some examples of hystereses could be identified exactly, the above mentioned problems of the original LOLIMOT algorithm appeared also in this case. Furthermore, a reasonable coupling of differential-dynamical and hysteresis models appeared to be difficult. Generally, the usage of state space models (with hidden states) is not possible with the existing algorithms. For these reasons, a further development or even complete rearrangement of model and algorithm seemed necessary.

Statistical decision theory (chapter 3)

Decision theory is a branch of probability theory and deals with the question which decisions can be made under uncertainty and how reliable these decisions are (an introduction can be found in Berger [1980] or, newer, in Robert [2001]). Decision theory is closely connected with the Bayesian interpretation of probability theory. The Bayesian interpretation of probability theory (in some way represented already by Laplace, later by Ramsey, de Finetti and Savage) led in the past (and is still leading) to quarrels between the "Bayesians" and the "Frequentists" (the latter ones represented by Fisher, Pearson, Neyman).

One main point criticized on the Bayesian approach is the necessity to introduce a-priori knowledge. But this actually must be seen as an advantage rather than as a disadvantage. The frequentist viewpoint, which consists of the opinion that a statistical analysis must be based on the pure data and not on a possibly subjective prior knowledge, leads often to senseless assumptions and unnecessary conservative and even wrong estimates (examples can be found in Jaynes [1976]). Contributing to this, in frequentist probability, statements are only allowed to be done for random experiments which are principally repeatable arbitrarily often in the same way. The Bayesian approach is much more flexible: to each variable (belonging to a random experiment or not), a distribution can be assigned. For this reason virtually every data set is useful (it does not need to be a random sample of a random experiment and also does not need to be "sufficiently large").

In the course of the mentioned quarrels, trials were undertaken to show Bayesian probability theory as the only reasonable theory serving to the modelling of uncertainty (known as Cox's theorem, see e.g. Jaynes [1990] and Arnborg and Sjödin [2003]). Despite some gaps in the original proofs it seems that this task has been treated successfully, albeit the definition of "reasonable theory" always stays to be somewhat arbitrary. In this sense the fuzzy-logic drops out as an alternative theory. Indeed, the fuzzy-logic provides a plentitude of possibilities (so, there exist several different fuzzy-logics), but does not provide any means to decide which of

these logics should be used under which circumstances. In contrast, if one accepts the above mentioned "reasonable" assumptions, then there is only one possible answer: the logic as induced by the Bayesian probability theory.

In system identification, parameter estimation in state space models can be done with Bayesian methods. A sever problem occurring is that apart from discrete or normal linear state space models, an analytic formula for the estimation of the hidden states is not available. Even in linear normal models, the joint estimation of hidden states and parameters is difficult. An estimate can only be approximated by numerical methods, and promising candidates for such methods are the nowadays investigated Sequential Monte Carlo (SMC) methods (sometimes also called particle filters; see e.g. Künsch [2005]).

Approximation theory (chapter 4)

Approximation theory (see e.g. DeVore [1998]) investigates the possibilities for the approximation of a complicated real function by other simpler real functions called approximants, and their convergence properties. The questions arising are:

- Which functions are suited to be used as approximants?
- Which functions can be approximated therewith?
- How good is this approximation?

Nonlinear approximation theory (the approximants are not taken from a vector space but from a manifold) is represented in the one-dimensional case by at least two classes of approximants: splines (with free knots) and wavelets. In the one-dimensional case both approaches lead to equivalent conclusions. In higher dimensions (the approximated function is multidimensional) this analogy breaks down, and it appears that the theory is no longer applicable to splines. Concerning wavelets, the theory is transposed to higher dimensions without problems. Thus, the question for the right approximants seems to be decided in favour of the wavelets. The other two questions mentioned, which functions can be approximated and how well this is done (i.e. how fast is the convergence?), can be answered by the theory of Besov spaces. These spaces can be obtained as interpolations of Sobolev spaces and encompass large classes of functions. Through the combination of these three theories, namely approximation, wavelet, and Besov space theory, which at first look do not seem to have something in common and which have developed independently, a powerful and flexible tool is gained for the approximation and analysis of finite-dimensional real functions. A similar theory for neural networks used as approximants does not exist, and some negative results indicate that such strong propositions as they exist for wavelets are not to be expected (see e.g. Hush [1999]). This is even more true for the approximation by means of fuzzy-logic: here not even an appropriate algorithm exists.

We mentioned above that three theories play a major rôle in this thesis: theory of nonlinear dynamical models and hysteresis, statistical decision theory based on Bayesian probability theory, and approximation theory in connection with wavelets. A combination of the first two theories may be done with the Sequential Monte Carlo methods. On the other hand, the

combination of Bayesian probability theory and wavelets was successfully applied in the field of image processing, both to the compression of (real or artificial) images and to the removal of noise (denoising); see Abramovich et al. [1998]. The crucial point can be found in the fact that the same technique which is used in function approximation with wavelets is also applied to the compression and denoising of images: the so-called thresholding. The effectiveness of this technique is guaranteed by two important properties of wavelets:

- the sparsity of the wavelet coefficient matrix and
- the decorrelation of the wavelet coefficients.

Exactly these properties lead to a new model class with corresponding identification algorithm.

The third possible combination of the above mentioned three theories essential for the thesis is the usage of wavelets for system identification. This is astonishingly a seldom appearing combination in the literature. For example, Hasiewicz [2001] uses wavelets for the identification of Hammerstein models. This model type is a simple form of nonlinear modelling, done by a serial connection of a nonlinear static and a linear dynamical model. The static part in this case is realized by a wavelet approximation whereas the linear dynamical part is treated with the usual identification procedures of linear systems theory. A complete different approach is given by the Unified Transform (UT) as presented in Feuer et al. [2006], a generalization of the Laplace, Laguerre, Hambo (Heuberger et al. [2003]), and other transforms given for Linear Time Invariant (LTI) systems. The Unified Transform shows properties closely related to the wavelet transform. It is however only applicable to linear systems. Nevertheless, a combination with the here presented methods could be interesting.

The new model class (chapter 5)

The new model class will be derived from the local model networks through the following modifications:

- Inclusion of non-Gaussian noise sources;
- Allowance of internal nonlinear differential dynamics represented by multi-dimensional real functions;
- Introduction of internal hysteresis models through two-dimensional “primitive functions”;
- Replacement respectively approximation of the weight functions and of the mentioned multi-dimensional functions by wavelets;
- Usage of the sparseness of the matrix of the wavelet coefficients;
- Identification of the wavelet coefficients with SMC methods.

The justification of these rearrangements is based upon results of the above mentioned theories. The new model class enables the identification of an essentially larger class of systems. Of course, these systems can further on have differential-dynamical properties, but also hysteresis properties and hidden states may now be incorporated. This enhanced flexibility is payed on the other side by an increased effort necessary for the identification. Once identified, the models are extremely fast during the simulation: The models resulting after the identification of the parameters consist mainly of recurrent linear filter banks (the same filter banks used with the fast wavelet transform FWT, which is known to be faster than the fast Fourier transform FFT).

Notations

$l := r$: definition of the left hand side l by the right hand side r

$\#A$: number of elements of a finite set A

$\complement A$: complement of the set A (in a larger set Ω)

$A \dot\cup B$: disjoint union of the sets A and B

$\mathbb{N} := \{0,1,2,3,\dots\}$ set of natural numbers including 0

$\mathbb{N}^* := \{1,2,3,\dots\}$ set of natural numbers excluding 0

$\mathbb{R}$: set of real numbers

$\mathbb{R}_{\geq 0} := \{x \in \mathbb{R} \mid x > 0\}$: set of positive real numbers

$\mathbb{R}_{>0} := \{x \in \mathbb{R} \mid x \geq 0\}$: set of non-negative real numbers

$\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$: set of real numbers including negative and positive infinity

$\mathbb{C}$: set of complex numbers

$\overline{g}(x) := \overline{g(x)}$: complex conjugate

$\ell^2(\mathbb{Z})$: square summable sequences over $\mathbb{Z}$

$L^2(\mathbb{R})$: square integrable functions on $\mathbb{R}$

$L^2(\Omega,\mu)$: square integrable functions on Ω with respect to a measure μ (defined on a σ-algebra $\mathscr{A}$ on Ω)

$\langle f,g\rangle := \int f(x)\overline{g}(x)dx$: scalar product on L^2

$L^p_{\mathrm{loc}}(T,X)$: locally p-integrable functions from T to X

$C^0(\mathbb{R}^d)$: set of all continuous and bounded functions $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ (or $\mathbb{C}$)

$S^\top$: transpose of the matrix S

$\mathbf{1}_A(x) := \begin{cases} 1, & \text{if } x \in A, \\ 0, & \text{else:} \end{cases}$ characteristic function of the set A

$\Pr(A|E)$: probability of the event A given the event E

$x \sim p(x)$: x is distributed according to $p(x)$

$g(x) \propto h(x)$: $g(x)$ is proportional to $h(x)$

$\mathbf{E}^p[h(x)] := \int h(x)p(x)dx$: expectation of $h(x)$ under the density $p(x)$

$\mathbf{E}_\theta[h(y)]$: expectation of $h(y)$ under the distribution $y \sim f(y\,|\,\theta)$

$\mathbf{E}^\pi[h(\theta)\,|\,y]$: expectation of $h(\theta)$ under the posterior distribution of θ, $\pi(\theta\,|\,y)$, given the prior π

$x_{1:n}$: $x_1, \ldots, x_n$ or $(x_1, \ldots, x_n)$

$A \asymp B$: there are constants $C_1, C_2 > 0$ such that $C_1 A \leq B \leq C_2 A$

1 Introduction: Grey-box models and the LOLIMOT algorithm

This chapter wants to give a first introduction into the main topics of this thesis. These topics are centred around the LOLIMOT algorithm, a simple and heuristic identification algorithm for local model networks. Local model networks will be described as a generalization of the concept of linear combinations of basis functions. Some of the topics presented in a rather informal way in the present chapter will be revisited in the subsequent chapters, where precise mathematical formulations and generalizations follow: dynamical systems, nonlinearity, parameter identification, atomic decompositions (of which local model networks are a special case), approximation theory.

Nevertheless, some fundamental notions of graph theory and especially a precise definition of decision trees will be given already in the present chapter. The structure of decision trees allows for the construction of normalized weight functions which can replace the weight functions originally used in the LOLIMOT algorithm. We consider also the application of a gradient based optimization method to locally optimize the parameters obtained by the LOLIMOT algorithm. This method enables some further improvements of the LOLIMOT algorithm, like the ability to use basis models other than ARX models, more flexible partitions of the regime space, and a pruning method.

Overview In the first section we describe (exemplified through a simple shock absorber) the relations between real systems and their models, focussing on several model schemes like white-, black- and grey-box models. We treat local model networks with the LOLIMOT algorithm as special nonlinear models with corresponding non-parametric identification algorithm. We then describe problems arising with this identification scheme and possible improvements. Some of these improvements that are easily implementable will be presented in greater detail.

Contributions

- Analysis of the problems occurring with LOLIMOT algorithm;
- Simple improvements given by incorporation of decision-tree based weight functions and gradient based local optimization algorithm;
- Further improvements based on the previous ones: incorporation of NOE model structure, and pruning.

1.1 Systems and models

1.1.1 Nonlinear dynamical systems and model schemes

Shock absorber as example of a nonlinear system

Looking inside We take a look inside a shock absorber as it is used in every car (see figure 1.1). We recognize a wide variety of different parts with different physical properties. The main part consists of a cylinder which surrounds a movable piston. Moved by a shock from outside, the piston presses oil inside the cylinder through wholes in the wall, thus dampening oscillations. A spring pushes the piston back to its original position, and a rubber stopper prevents the piston from plugging against the walls of the cylinder when shocks are too strong. Thus, the shock absorber comprises the interaction of the mechanical movements of rigid bodies, the visco-elastic dynamics of fluids, the elastic behaviour of spring-damper systems, the deformations of elasto-plastic materials, etc. If one wanted to simulate all these effects based on the equations representing the physical laws that govern the individual parts, a complicated coupling of solvers for algebraic differential equations, ordinary nonlinear differential equations and partial differential equations would be needed. The computational complexity would be very high.

Looking from outside In contrast, looking from outside onto the shock absorber, we only are aware of the phenomenological properties. We observe aspects like nonlinear stiffness, like nonlinear viscous damping when the shock absorber is excited with high frequencies, like hysteretic effects when excited with low frequencies, but we are not able to assign these phenomena to the individual parts of the shock absorber. There exist also classes of mathematical equations describing these more phenomenological properties of the shock absorber. The knowledge in this case is of a more qualitative nature. Experiments with corresponding measurements have to be done to gain the necessary quantitative information needed to be able to decide for the right equations and exact parameters.

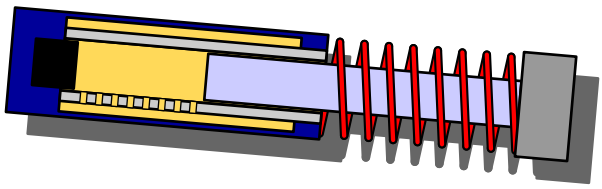


Figure 1.1: Schematic view inside a shock absorber

Nonlinear systems and adequate models Nonlinear dynamical systems like the shock absorber play an important role in technical, biological, medical, and social life and sciences. In all these disciplines where individual parts interact in time and space in a complex way, one tries to build models of the systems to get insight into them. The complexity of these systems forces one to use computers for simulation, analysis, and control of the systems, and this in turn cannot be done without adequate models of the given real systems. This adequacy of

the models can be seen under different aspects. Thus, the main interest may be in exactness or interpretability of the model, in the necessity of simple and fast computations, in models which are robust against uncertainties concerning the system or the model, in the ability to use the models to make prognoses of future behaviour or in the applicability as underlying basis for control tasks.

When dealing with simulations of reality, we are concerned with at least two "systems": the ***real system***, and the mathematical system reflecting the behaviour of the real system, called the ***model*** of the real system. The real system is what we find in reality. It is governed by the laws of nature, may it be a physical, chemical, biological or social system. The model tries to mimic the behaviour of the real system. The model is represented as a set of mathematical equations which we believe describe the laws of nature, at least up to an adequate approximation. To test our believes, we need to apply experiments on the real system and to compare the results to the results predicted by the model.

Information relating real system and model Thus, our information about the real system in relation to its model consists of three kinds: ***prior knowledge*** about the real system like physical phenomena or structure, ***observed data*** driven from experiments, and ***assumptions*** on the adequacy of our model. We return to these three kinds of information in more detail in chapter 3.

We want to take for granted that our assumptions on the relations between real system and model are correct. But how can we incorporate the other two types of information into our model, the prior knowledge and the data? These two types of information lead to two different ways of modelling.

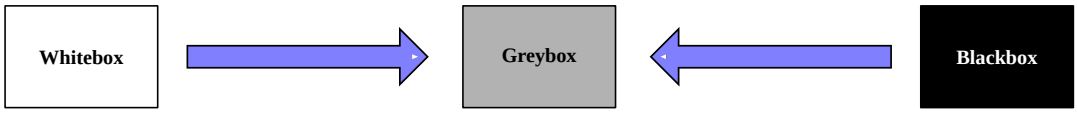
White-box models If we model the a-priori knowledge about the structure of the system, we try to directly translate this knowledge into mathematical equations. Ideally, these equations already give the model. Since we have a complete insight into the way our system works and because we use this to build the model, this kind of model is called a ***white-box model***. White-box models try to make a more or less exact image of the physical laws and behaviour of the given real system. Model structures belonging into this category are ordinary or partial differential equations and their discrete counterparts the difference equations, as well as differential algebraic equations (used in multi-body systems); also expert systems or fuzzy systems may be mentioned.

Black-box models In contrast, when using measurements gained by experiments on the real system to build the model, these measurements alone do not give us any insight into the real system, and also no understanding of how the system works is brought into the construction of the model. Therefore, this type of model is called a ***black-box model***. Black-box models are mainly provided by weighted combinations of general or specialized basis functions (prominent examples are artificial neural networks). Generally speaking, we have to choose out of an often parameterized set of possible models that model which fits optimally to the measured data. We are concerned with an optimization problem: The parameters (as e.g.

included in the basis functions or appearing through the weight functions) have to be identified using measured data taken from the real system.

Advantages and disadvantages The different model schemes have different advantages and disadvantages. The white-box structures provide exact models. But with increasing complexity of the real system also the complexity of the model increases and with this the efforts in time and costs for both implementing and running the model. On the other side, the (non-interpretable) black-box models often need by far less of the mentioned efforts. The disadvantage of this model scheme is the difficulty to introduce many kinds of phenomenological or qualitative knowledge one surely has got about the real system. This may lead to badly identified parameters. One should note that each experiment can reveal only a small aspect of the behaviour of a complex system. Our measurements alone are never sufficient to build a model. We always need to add some more of our knowledge about the system.

Grey-box models Of course, the notions of white- and black-box models are idealistic ones. In reality, modelling is always something in between these two extremal views: Even if we have some structural insight into our real system and we want to apply white-box modelling, we often do not know the exact parameters in the derived equations. These parameters have to be identified via measurements and optimization. Or, reversibly, the set of possible models for black-box modelling is often guided by some knowledge about the phenomenological behaviour of the given system. In all cases, we are actually using a model type called ***grey-box model***, which constitutes a mixture between white- and black-box model schemes (see figure 1.2).



White-box Modelling:

- Physical knowledge used
- Differential equations, Fuzzy Systems
- +: very high interpretability
- −: needs exact knowledge of system

Black-box Modelling:

- Measured data used
- Parameter identification, Neural Nets
- +: Simulation very fast
- −: not interpretable

Figure 1.2: White-box and Black-box models compared

Examples of grey-box models are neuro-fuzzy systems or semi-physical models (e.g. Sjöberg [2000], Lindskog and Ljung [1995]).

Linearity and nonlinearity A nice property of systems is ***linearity***. The characterizing property of a ***linear system*** is that:

(i) if the system is excited with input u_1 resulting in an output y_1, and on the other side with input u_2 resulting in the output y_2, then the superposed input $u_1 + u_2$ will result in the superposed output $y_1 + y_2$; and

(ii) if the system is excited with the input u resulting in an output y, then the scaled input λu will result in the scaled output λy.

Summarized, a superposition of several scaled inputs will result in a corresponding superposition of scaled outputs. If this property is not necessarily fulfilled, the system is called a ***nonlinear system***.

In real systems, linearity is often only approximately given in certain ranges of excitements of the system. An example may be a spring. If a relatively small force is applied to the spring by pulling on a loose end while the other end is fixed, the displacement of the former end is in a fairly linear relationship to the applied force; but if the force increases, the displacement will be bounded to some maximal position, and a kind of saturation effect occurs: this is a nonlinear behaviour. Nevertheless, this effect may still be described by a nonlinear differential equation. This is not anymore possible if the force even more increases: then the spring looses its property of being elastic, and irreversible plastic deformations show an even more nonlinear effect. Such plastic effects are common to materials like rubber, and lead at the end to long-memory effects called hysteresis which cannot be captured by ordinary differential equations.

The aim of this thesis Grey-box models provide the possibility to include both physical knowledge for structural modelling and data knowledge for the choice, identification and adaptation of the included parameters. Our aim in this thesis is to answer at least partly the question: How is it possible to build models out of structural or phenomenological knowledge together with measured data driven from experiments in a systematic way? The provision of model types for grey-box modelling of nonlinear systems together with appropriate identification algorithms is the aim of this thesis. Our focus will be on phenomenological models for nonlinear dynamical systems. Therefore we will provide model classes for several phenomena (such as nonlinear differential dynamics or hysteresis) together with algorithms which allow to choose in some way an optimal or nearly optimal model which fits to given measurements.

What is needed? Generally speaking, at first we need a mathematical formalization (model or model scheme) of dynamical systems which is able to describe several nonlinear phenomena like:

- nonlinear differential dynamics,
- hysteresis effects,
- etc.

We need secondly an identification algorithm which adapts the model to data (measurements) obtained from a given real system through experiments (identification). At last, we also need an implementation into a computer programme.

Shock absorber revisited Returning to the shock absorber example, we are aware that modelling the physics is very complicated and time consuming. Whereas white-box models are too complicated, black-box models are too restricted concerning the incorporation of phenomenological aspects. Halfway between both of them we locate the grey-box models: simpler models based on more phenomenological considerations which can easily be adapted after measurements are available.

1.1.2 Properties of systems and models

Deterministic and stochastic models

The identification of a model by measured data is complicated by the problem that measured data are always disturbed by errors. The sources of these errors are different. We have:

- Measurement errors due to restrictions of measurement equipment,
- Model errors due to unmodelled phenomena (like aging etc.),
- Noise originating from outside or inside the system,
- etc.

Therefore, each identification algorithm must be robust against data errors. To deal with disturbed data and errors, we must include these errors into our models. One can find two possibilities to do this:

- **Deterministic:** We assume that our data are given by a function which is affected by some (usually bounded) function.
- **Stochastic:** We assume that our data are given by a stochastic process equipped with a probability distribution.

In this chapter we specialize on systems which can be described by a finite superposition of linear difference equations. There is no hidden dynamics, i.e. all quantities determining the dynamical behaviour of the systems are observable from outside. Furthermore, all noises are assumed to be Gaussian (or simply neglected). This allows us to use a relatively simple algorithm. In the following chapters, we will subsequently widen our systems and models.

Static and dynamical systems

When is a system a dynamical system? What makes the difference to static systems? Shortly said: Dynamical systems evolve in time. Mathematically, (a model of) the time may be given by values t out of a subset $\mathscr{T}$ of $\mathbb{R}$. Let further be given a set $\mathscr{U}$ of inputs $u \in \mathscr{U}$ and a set $\mathscr{Y}$ of outputs $y \in \mathscr{Y}$. A stochastic model for a ***static system*** is given by a probability measure on a σ-algebra of $\mathscr{Y}$ conditioned on the input $u \in \mathscr{U}$. This reduces to a simple function $f : \mathscr{U} \longrightarrow \mathscr{Y}$ defined on the input domain $\mathscr{U}$ with values in the output domain $\mathscr{Y}$ in the deterministic case. If $\mathscr{U}$ and $\mathscr{Y}$ themselves depend on the time $\mathscr{T}$, and if for some times t_1 and t_2 the

corresponding inputs $u(t_1)$ and $u(t_2)$ are equal, $u(t_1) = u(t_2)$, then the corresponding outputs $y(t_1)$ and $y(t_2)$ are equally distributed, and in the deterministic case thus equal, $y(t_1) = y(t_2)$. For ***dynamical systems***, this may not be true. Here, we have to use stochastic processes, or, in the deterministic case, an operator Γ to describe the system. For now, we want to consider dynamical systems given by

$$y(t) := \Gamma(u(\cdot))(t) + V(t) \qquad \text{for all } t \in \mathscr{T} \subseteq \mathbb{R}$$

where $V(t)$ is the noise process which is usually assumed to be „small“ in some sense. Thus, the output $y(t)$ at time t does not depend on the input $u(t)$ at time t only, but on the whole function $u : \mathscr{T} \longrightarrow \mathscr{U}$, which we sometimes denote by $u(\cdot)$ if we want to emphasize that it is a function. For real systems, we surely expect that at a time t the system output $y(t)$ does actually only depend on inputs $u(\tau)$ in the past and the present, $\tau \leq t$, but not in the future $\tau > t$. This causality aspect and some others necessary to define dynamical systems in a plausible way will be treated axiomatically in chapter 2 for the deterministic case, and in chapter 3 for the general stochastic case. There, we will give a definition of dynamical systems covering a wide variety of real and theoretical systems evolving in time. For now, we will be content with the sloppy definition given above by the input-output operator Γ and the noise process V. In this chapter we always assume $\mathscr{U}$ to be the m-dimensional Euclidean space $\mathbb{R}^m$, $m \in \mathbb{N}$, and similarly $\mathscr{Y}$ to be the d-dimensional Euclidean space $\mathbb{R}^d$.

Time-continuous and time-discrete models

One distinguishes between models where the time domain $\mathscr{T}$ is $\mathbb{R}$ (or an interval on $\mathbb{R}$), called ***(time)-continuous models***, or models where $\mathscr{T} = \mathbb{Z}$ (or an interval of $\mathbb{Z}$), called ***(time)-discrete models***. Since observations are anyway almost always given only for some discrete time points, it is usually enough to consider time-discrete models.

Predictors and error function

Our aim is to construct a model which fits best to the real system. Often, this fitting is measured by means of some kind of ***error function*** $e : \mathscr{M} \longrightarrow \mathbb{R}$ where $\mathscr{M}$ is some model class. The simplest possibility is to use some norm on the difference between measured output y of the real system and some predicted output $\hat{y}$ of a given model Σ:

$$e(\Sigma) := \|y(\cdot) - \hat{y}(\cdot)\| \qquad \text{for all } \Sigma \in \mathscr{M}.$$

For example, if the predictor is given by the operator Γ, i.e.

$$\hat{y}(t) = (\Gamma u(\cdot))(t),$$

and the data y are realizations of $\Gamma(u(\cdot))(t) + V(t)$, then the difference $y(t) - \hat{y}(t)$ is just a realization of the noise process $V(t)$.

If the right model is unknown and we are looking for the best model in $\mathscr{M}$ fitting to the data, we may choose a model Σ out of the model class $\mathscr{M}$ which minimizes the error function e. In most cases, this set $\mathscr{M}$ of proposal models is given by a parameterized family of models. In the next subsection we show some examples of how to obtain such model classes.

1.1.3 Separation of dynamics

Inner and outer dynamics

As mentioned earlier, the measured output $y(t)$ of a dynamical system or the predicted output $\hat{y}(t)$ of the model at time t does not only depend on the input $u(t)$ at the actual time t but also on previous inputs $u(\tau)$ for $\tau \leq t$, and the system is described by means of an operator Γ with

$$\hat{y}(t) = (\Gamma u(\cdot))(t).$$

To get a better grip on such an operator, we decompose it into the (dynamical) ***state transition operator*** φ and a (static) ***output function*** η:

$$\Gamma := \eta \circ \varphi \qquad \text{with} \qquad x(t) := \varphi(u(\cdot))(t) \qquad \text{and} \qquad \hat{y}(t) := \eta(x(t)),$$

thus introducing ***(internal) states*** x taken from some ***state domain*** $\mathscr{X}$. In this chapter we also assume $\mathscr{X}$ to be equal to some Euclidean space $\mathbb{R}^n$. It is of course always possible to decompose an operator Γ in this way, one just has to choose $\mathscr{X} = \mathscr{Y}$ and η the identity function. But for a model to be a ***state space model*** or ***state space system***, one requires additionally that the state $x(t)$ at a time t has accumulated all information from the history of the system necessary to determine the output of the system as well as all future states. Then the choice $\mathscr{X} = \mathscr{Y}$ is in most cases not justified. We distinguish two cases: Either the states $x(t)$ depend exclusively on past and present inputs $u(\tau)$, $\tau \leq t$, on past outputs $y(\tau)$, $\tau < t$, of the real system, and on past outputs $\hat{y}(t)$, $\tau < t$, of the model via a multi-dimensional function; then we say the model exhibits ***outer dynamics***. Or else the states are not observable from outside; then we say the model exhibits ***inner dynamics*** and the states are ***hidden***.

State transitions In the general case where $\mathscr{X}$ is different from $\mathscr{Y}$ with states which are not visible from outside, one often gains enough flexibility that also the state transition operator can be defined by means of a static recursive function. In this case, the state $x(t+1)$ at time $t+1$ exclusively depends on the state $x(t)$ at time t and on the input $u(t)$ at time t. The operator φ is thus given through a ***state transition function*** $\psi : \mathscr{X} \times \mathscr{U} \longrightarrow \mathscr{X}$ and an ***initial state*** $x_0 \in \mathscr{X}$ at some initial time $t_0 \in \mathscr{T}$. We are then able to compute $x(t)$ recursively, at least for all $t \geq t_0$. In the case of discrete models showing outer dynamics where the states consist of a *finite* number of past inputs and outputs, this is always possible.

Difference dynamics Outer dynamics in the discrete case is usually given by ***difference dynamics***, also called ***tapped delays***. Here the components of the state vector $x \in \mathscr{X} = \mathbb{R}^n$ are delayed inputs and outputs; if the outputs are the outputs of the real system (measurements), the states are defined as

$$x(t) = \big(u(t), u(t-1), \ldots, u(t-n_u), y(t-1), \ldots, y(t-n_y)\big)^\top.$$

The overall model (with noise process) is then called ***NARX*** (**N**onlinear **A**uto**R**egressive with e**X**ternal input) model or ***nonlinear equation error model***. If we instead take the delayed model outputs, we get

$$x(t) = \big(u(t), u(t-1), \ldots, u(t-n_u), \hat{y}(t-1), \ldots, \hat{y}(t-n_y)\big)^\top,$$

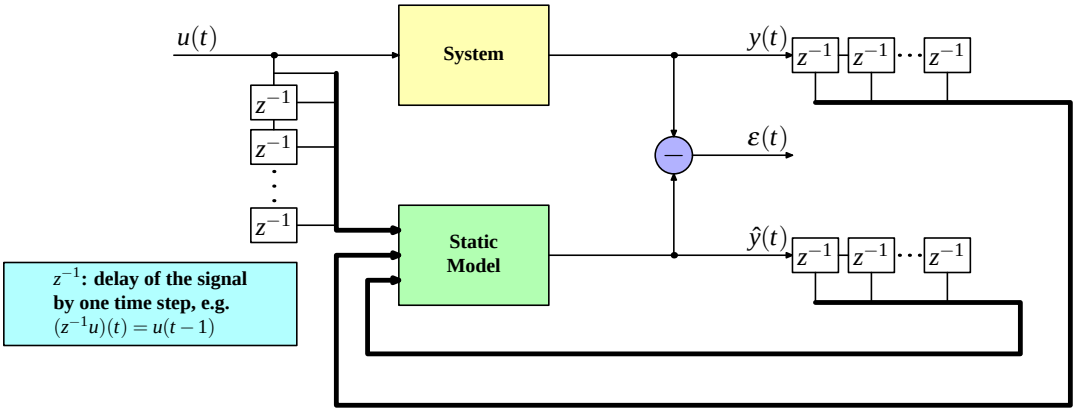


Figure 1.3: Tapped Delays

which yields the so-called ***NOE*** (**N**onlinear **O**utput **E**rror) model (see figure 1.3). In both cases the natural numbers n_u and n_y denote the maximal number of steps we have access to of the input respectively output values in the past.

Simulation There is a fundamental difference between these two model types concerning the inputs into the model. We defined the input into the system as functions $u(\cdot) : \mathscr{T} \longrightarrow \mathscr{U}$. For the NARX model, this is not correct. When looking at the definition of x, we see that we need the actual inputs $u(t), u(t-1), \ldots, u(t-n_u)$, but we need as well the measured(!) outputs $y(t-1), \ldots, y(t-n_y)$. So the real input into the system is defined by functions over time with values given by pairs $(u,y) \in \mathscr{U} \times \mathscr{Y}$:

$$\hat{y}(\cdot) := \Gamma^{\mathrm{NARX}}(u(\cdot), y(\cdot)).$$

In the case of the NOE model, the measured outputs are not needed, and we simply have

$$\hat{y}(\cdot) := \Gamma^{\mathrm{NOE}}(u(\cdot)).$$

Thus, if we want to *simulate* the model without recourse to measured output data, we need to replace all occurring measured outputs y by the ***simulated outputs*** $\hat{y}_u$ (depending on u alone, and not on y). In the case of the NARX model, we have thus obviously $\hat{y} \neq \hat{y}_u$, whereas in the NOE model $\hat{y} = \hat{y}_u$. Since the model choice (parameter identification) is done with the predictor $\hat{y}$ (using measured output data), it is probable that the use of a NOE model will lead to better simulation results than the use of an ARX model.

Initial values We still need some more information: initial values for the state x at the initial time t_0. If we start our computations at time t_0, the values $u(t), u(t-1), \ldots, u(t-n_u)$ and $y(t-1), \ldots, y(t-n_y)$ may not be defined, and the values $\hat{y}(t-1), \ldots, \hat{y}(t-n_y)$ are surely not defined. For this reason, we have to fix them in advance and put them as additional inputs into the model. In both cases, the initial state $x_0 = x(t_0)$ looks like:

$$x_0 = \left(u_{t_0}, u_{t_0-1}, \ldots, u_{t_0-n_u}, y_{t_0-1}, \ldots, y_{t_0-n_y}\right)^\top$$

with

$$u_{t_0}, u_{t_0-1}, \ldots, u_{t_0-n_u} \in \mathscr{U}, \qquad y_{t_0-1}, \ldots, y_{t_0-n_y} \in \mathscr{Y}.$$

The predicted/simulated output is then defined for all times $t \geq t_0$. We denote the set formed by these times by $\mathscr{T}_{t_0}$:

$$\mathscr{T}_{t_0} := \left\{ t \in \mathscr{T} \;\middle|\; t \geq t_0 \right\} \qquad \text{for some } t_0 \in \mathscr{T}.$$

When depending on the initial time t_0 and initial state x_0, we write the operator Γ as Γ_{t_0,x_0}. The predicted output $\hat{y}(\cdot) = \Gamma_{t_0,x_0}(u(\cdot))$ is then a map $\hat{y}(\cdot) : \mathscr{T}_{t_0} \longrightarrow \mathscr{Y}$, or shorter $\hat{y}(\cdot) \in \mathscr{Y}^{\mathscr{T}_{t_0}}$. (One actually needs only input functions $u(\cdot) \in \mathscr{U}^{\mathscr{T}_{t_0}}$ and output functions $y(\cdot) \in \mathscr{Y}^{\mathscr{T}_{t_0}}$.)

Linear models Let now $\mathscr{U} := \mathbb{R}^m$ and $\mathscr{Y} := \mathbb{R}^d$. Linearity of the NARX models (in this case one simply calls them ARX models) is then given when for $d = m = 1$

$$\eta(x(t)) = A\big(u(t), u(t-1), \ldots, u(t-n_u)\big) + B\big(y(t-1), \ldots, y(t-n_y)\big)^\top$$

with matrices $A \in \mathbb{R}^{d \times n_u+1}$ and $B \in \mathbb{R}^{d \times n_y}$. In the general case $d \geq 1$ and $m \geq 1$, we need m matrices $A_i \in \mathbb{R}^{d \times n_u+1}$, $i = 1, \ldots, m$, and d matrices $B_j \in \mathbb{R}^{d \times n_y}$, $j = 1, \ldots, d$. Then:

$$\eta(x(t)) = \sum_{i=1}^{m} A_i\big(u_i(t), u_i(t-1), \ldots, u_i(t-n_u)\big) + \sum_{j=1}^{d} B_j\big(y_j(t-1), \ldots, y_j(t-n_y)\big)^\top,$$

$u_i(\tau)$ and $y_j(\tau)$ being the components of the vectors $u(\tau)$ and $y(\tau)$, respectively. All these components can be rearranged such that x is a column vector of dimension

$$n := m \cdot (n_u + 1) + d \cdot n_y,$$

and the coefficients of the A_i and B_j can be gathered into a $d \times n$ matrix θ. We thus can simply write

$$\hat{y}(t) := \eta(x(t)) = \theta x(t).$$

The case for the linear NOE model (called OE model) is similar (replace y by $\hat{y}$).

Both model types have a special property: they are ***linear*** on the inputs. If we have two input functions $u^{(1)}(\cdot), u^{(2)}(\cdot) \in \mathscr{U} = \mathbb{R}^m$, and a scalar $\lambda \in \mathbb{R}$, then the sum $u^{(1)}(\cdot) + u^{(2)}(\cdot)$ as well as the product $\lambda u^{(1)}(\cdot)$ are again in $\mathscr{U}$, and the linearity property requires the following equations to hold:

$$\begin{aligned} \Gamma\big(u_1(\cdot) + u_2(\cdot)\big) &= \Gamma\big(u_1(\cdot)\big) + \Gamma\big(u_2(\cdot)\big), \\ \Gamma\big(\lambda u_1(\cdot)\big) &= \lambda \Gamma\big(u_1(\cdot)\big). \end{aligned}$$

This does not mean anything else but that the input-output operator Γ is a linear operator. The linearity for the ARX and OE models is easily established.

There exists a broad and well developped theory for the identification of continuous and discrete linear models (see e.g. Ljung [1999]). We are interested in nonlinear systems. In the next section, we describe how nonlinear systems may be built with linear building blocks, or,

reversing the direction, how nonlinear systems can be decomposed into these linear building blocks.

The ARX model has an additional linearity. Here, also the model is linear in the parameters. This is not true for the OE model, because in the formula $\theta x(t)$ the $x(t)$ depends also on the parameter θ (remember that $x(t)$ contains past model outputs which obviously depend on the parameters of the model!). The linearity in the parameters of the ARX model makes the identification of these parameters extremely easy. Only linear regression has to be applied. The disadvantage of this model is the need for measured outputs of the real system. This may be available in some control settings, but for applications where simulation of the real system is needed without having access to measurements during the simulation run, this model is not adequate. Here, the OE model must be used. But identification of the parameters is more involved and can be done only using iterative algorithms.

1.1.4 Linear combinations of basis functions and networks

Linear combinations of basis functions We follow Sjöberg et al. [1995]. We consider the case where the output map $\eta(x)$ is nonlinear. In this case, a usual method is to approximate η by a ***linear combination of basis functions*** η_k:

$$\eta(x) \approx \sum_k \alpha_k \eta_k(x).$$

These basis functions η_k do not necessarily have to be functional bases (like orthonormal bases of a Hilbert space). The key is to determine the basis functions η_k. In most cases, they are derived from a ***mother basis function*** κ through translations and dilations:

$$\eta_k(x) := \kappa(x;\beta_k,\gamma_k) = \kappa\Big(\frac{x-\gamma_k}{\beta_k}\Big),$$

where this equation has to be interpreted more or less symbolically. The ***scale parameter*** β_k refers thus to a scale or directional property of η_k, whereas the ***location parameter*** γ_k denotes a location or position of the η_k.

Basis functions in the scalar case Examples in the scalar (single-variable) case, i.e. $x \in \mathbb{R}$, are:

- Fourier series: $\kappa(x) = \cos(x)$; the corresponding linear combination is then the Fourier series expansion, with $1/\beta_k$ corresponding to the frequencies and γ_k to the phase.
- Piecewise constant functions: Take κ as the characteristic function of the unit interval:

  $$\kappa(x) := \begin{cases} 1 & \text{if } 0 \le x < 1, \\ 0 & \text{else.} \end{cases}$$

  Take further e.g. $\gamma_k := C_\Delta k$ and $\beta_k := C_\Delta$ with a constant $C_\Delta \in \mathbb{R}^+$. With $\alpha_k := \eta(C_\Delta k)$, the linear combination

  $$\eta(x) \approx \sum \alpha_k \kappa\Big(\frac{x-\gamma_k}{\beta_k}\Big)$$

yields then a piecewise continuous approximation of η. A similar result can be obtained by using a smooth version of the characteristic function, e.g.

$$\kappa(x) := \frac{1}{\sqrt{2\pi}} e^{-x^2/2},$$

the Gaussian bell function.

- A variant of the last example is the following: Take the unit step function

$$\kappa(x) := \begin{cases} 0 & \text{if } x < 0, \\ 1 & \text{if } x \geq 0. \end{cases}$$

This gives similar results as the previous example, because the characteristic function of the unit interval can be obtained by the difference of two step functions. A smooth version is the ***sigmoid function***

$$\kappa(x) := s(x) := \frac{1}{1+e^{-x}}$$

with similar results.

One can distinguish two classes of basis functions on a single variable; the discrimination is done according to the variability of the η_k's, given by the behaviour of the gradient

$$\eta_k'(x) := \frac{d\eta_k(x)}{dx}.$$

One has:

- ***Local basis functions***: their gradient η_k' has a bounded support or at least vanishes rapidly at infinity; thus, the variations are essentially bounded to some interval.
- ***Global basis functions***: their gradient η_k' has infinite support with values $\gg 0$ (these may be bounded or not).

The Fourier series provides global basis functions, whereas the other examples are based on local basis functions.

Basis functions in the multi-dimensional case In the multi-dimensional case ($x \in \mathbb{R}^n$ with $n > 1$), the basis functions η_k are multi-variable functions which are often derived from a single-variable mother basis function κ. The following constructions can be found:

- ***Tensor product***: Let be given n single-variable basis functions $\eta_k^{(1)}, \ldots, \eta_k^{(n)}$, then the multi-variable basis function η_k given by the tensor product construction is defined by the product

$$\eta_k(x) := \eta_k^{(1)}(x_1) \cdots \eta_k^{(n)}(x_n).$$

- ***Radial construction***: Let κ be an arbitrary single-variable function. Then the radial construction of multi-variable basis functions is given by

$$\eta_k(x) := \eta_k(x;\beta_k,\gamma_k) := \kappa(\|x-\gamma_k\|_{\beta_k}) \qquad \text{for } x \in \mathbb{R}^n$$

  where $\gamma_k \in \mathbb{R}^n$ and $\|\cdot\|_{\beta_k}$ is a norm depending on β_k, e.g.

$$\|x\|_{\beta_k} = \sqrt{x^\top \beta_k x}$$

  with β_k a positive definite matrix of scale parameters depending on k.

- ***Ridge construction***: Let κ be a single-variable function. Then for all $\beta_k \in \mathbb{R}^n$ and $\gamma_k \in \mathbb{R}$, a ***ridge function*** is given by

$$\eta_k(x) := \eta_k(x;\beta_k,\gamma_k) := \kappa(\beta_k^\top x + \gamma_k), \qquad \text{for } x \in \mathbb{R}^n.$$

Examples falling in the above mentioned categories are wavelets, sigmoidal or radial basis networks, kernel estimators, B-splines, hinging hyperplanes, projection pursuit regression, and even Fuzzy models (compare to Sjöberg et al. [1995] and Juditsky et al. [1995]). Several of them will be presented in later chapters.

Neural networks If we write the linear combination of basis functions in the slightly different form

$$\eta(x) = \sum_{k=1}^{n} \alpha_k \kappa(\beta_k^\top x + \gamma_k), \qquad \text{for } \alpha_k \in \mathbb{R},\ \beta_k,\gamma_k \in \mathbb{R}^n$$

(which is nevertheless equivalent to the previously given expression), we have exactly the equation of a ***multilayer network*** with one ***hidden layer***. The hidden layer, where "hidden" means "not accessible from outside", is given by the functions $\eta_k := \kappa(\beta_k x + \gamma_k)$. Accessible from outside is the input layer given by the input $x = (x_1,\ldots,x_n)^\top$, and the output layer, the value $\eta(x)$. We could now increase the number of layers by repeating the procedure of building linear combinations: Write $x^{(1)} := x$, $\beta_k^{(1)} := \beta_k$, and $\gamma_k^{(1)} := \gamma_k$, denote the outputs of the basis functions by

$$x_k^{(2)} := \kappa(\beta_k^{(1)\top} x^{(1)} + \gamma_k^{(1)}),$$

and collect them into a vector $x^{(2)} := (x_1^{(2)},\ldots,x_d^{(2)})^\top$. Now this value is taken as the input into the next layer, and so on,

$$x_k^{(i+1)} := \kappa(\beta_k^{(i)\top} x^{(i)} + \gamma_k^{(i)}), \qquad \text{for } \beta_k^{(i)},\gamma_k^{(i)} \in \mathbb{R}^n.$$

The basis functions

$$\kappa(x^{(i)};\beta_k^{(i)},\gamma_k^{(i)}) = \kappa(\beta_k^{(i)\top} x^{(i)} + \gamma_k^{(i)})$$

constitute thus the i-th hidden layer. The output layer of an N-hidden-layer network is finally given by

$$\eta(x) := \sum_{k=1}^{n} \alpha_k x_k^{(N)}, \qquad \text{for } \alpha_k \in \mathbb{R}.$$

Nevertheless, the most common multilayer networks contain only one hidden layer and are thus equivalent to linear combinations of basis functions.

Recurrent neural networks In contrast to feedforward neural networks where the connections are directed from the inputs to the outputs, ***recurrent neural networks*** have also feedback connections (loops) and show a dynamical behaviour: The backward transitions of the signals usually occur at the next time-step. If the recurrent network is represented as a feedforward network with additional feedback connections only going from the outputs to the inputs of this feedforward network, then this recurrent network is called to be in ***canonical form***. In this special case we are again in the situation of outer dynamics, as with the NARX and NOE models. The NOE model can be seen as a special kind of recurrent neural network in canonical form. In contrast to the NOE model, recurrent neural networks usually have hidden nodes which are not accessible from outside. It may also be that some nodes of the input and output layers of the feedforward network are not available from outside.

It should be noted that every recurrent neural network can be rearranged such that it is in canonical form (see [Nerrand et al., 1993]). These nets have then exactly the form of a state space model, where the state transition function φ is realized by the feedforward neural net. Such neural nets may also be obtained by semi-physical modelling, where some physical knowledge on the system is given by differential equations and the unknown parts are modelled as black box with neural networks. After discretization, the resulting difference equations can be interpreted as a recurrent neural network which afterwards can be rearranged into the canonical form (see Dreyfus and Idan [1998]).

1.2 Local model networks

Local model networks were developed in different fields with different names. Nelles [2001] mentions also the names Takagi-Sugeno fuzzy-models, operating regime based models, piecewise models and local regression, coming from disciplines like neural networks, fuzzy logic, statistics, and artificial intelligence, with close links to multiple model, mixture of experts, and gain scheduling approaches (see also Johansen and Murray-Smith [1997]).

Definition

A ***local model network (LMN)*** is a parallel superposition of partial models which become local models by localizing weights (see figure 1.4). Mathematically, a parallel superposition of the partial models is just given by summing up the outputs of these models. Since different partial models should be valid for different parts in the state space, weight functions are introduced which provide this localizing effect:

$$(\Gamma u)(t) := \sum_{k=1}^{N} \alpha_k(t)(\Gamma_k u)(t)$$

where the Γ_i are the local models and $\alpha_k : \mathscr{T} \longrightarrow \mathbb{R}$ are the weight functions. If we write the partial models in state space form with joint state transition function,

$$(\Gamma_k u)(t) = \eta_k(x(t)), \qquad k = 1,\ldots,N, \qquad x(t) = (\varphi u)(t),$$

we have

$$(\Gamma u)(t) = \sum_{k=1}^{N} \alpha_k(t)\eta_k(x(t)).$$

In this definition, the weights are time dependent. The usual approach is to restrict this dependency on the "active regime", i.e. one assumes also a decomposition

$$\alpha_k(t) = w_k(x(t)), \qquad k = 1,\dots,N.$$

In this case, the weight functions are called ***regime based weight functions***. The overall local model network in state space form with regime based weight functions is then

$$(\Gamma u)(t) = \sum_{k=1}^{N} w_k(x(t))\eta_k(x(t)).$$

We usually want to assume this form of local model networks. Using the terminology of artificial neural networks, the single local models are often called ***neurons*** of the local model network.

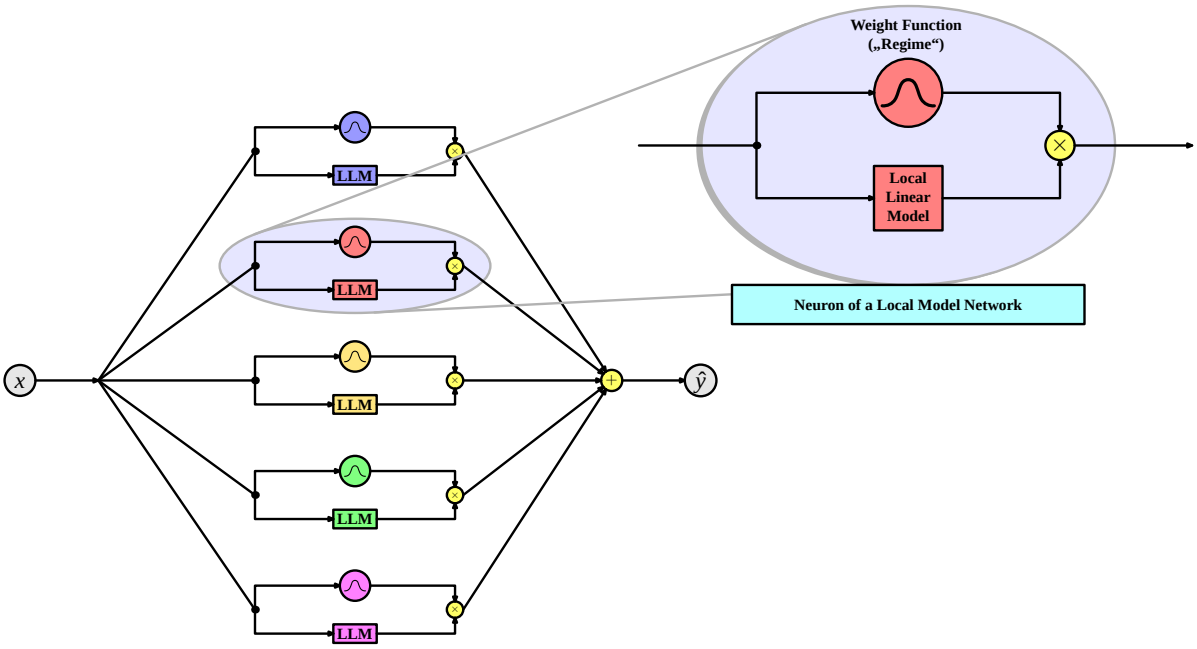


Figure 1.4: Local Model Network and Neuron

Weights To provide a better interpretability of local model networks, we put some conditions on the weight functions $w_k : \mathscr{X} \longrightarrow \mathbb{R}$:

$$\sum_{k=1}^{N} w_k(x) = 1 \qquad \text{and} \qquad w_k(x) \geq 0 \qquad \text{for all } x \in \mathscr{X}.$$

We call such weight functions ***normalized***. The local model network is thus a convex combination of the partial models.

Parameters Usually weight functions w_k and output functions η_k are given by parameterized families, i.e.:

$$w_k(x) := w\left(x; \theta_k^w\right) \qquad \text{and} \qquad \eta_k(x) := \eta\left(x; \theta_k^\eta\right)$$

where θ_k^w and θ_k^η are usually taken from Euclidean spaces. These parameters are a-priori unknown and must therefore be identified. For the identification process it is important to know how they are involved in the equations. Here lies the main difference between weight functions w_k and output functions η_k: The parameters θ_k^w of the weight functions are assumed to be nonlinear, whereas the parameters θ_k^η of the output functions are linear parameters.

Regression vector Under the assumption that the parameters θ_k^η are linear, one can decompose each output function into a scalar product of θ_k^η and a vector $h^\eta(x)$, called ***regression vector***. It is usually assumed that the function h^η is known and computable. We thus have the output functions

$$\eta\left(x; \theta_k^\eta\right) = {\theta_k^\eta}^\top h^\eta(x).$$

In this way, also e.g. polynomial models can be summarized under the term "linear model", then understood as linear with respect to the regression vector $h^\eta(x)$. It should be noted that with this interpretation, there is no difference any more between our models which are linear in the parameters and the usually so-called linear models: they are linear with respect to the regression vector $h^\eta(x)$.

Linear partial models If we take the function h^η to be the identity, then the output function is linear with respect to the states $x \in \mathscr{X}$. If additionally the state transition φ is also a linear operator, the partial models are called linear. Models with linear output function can be written in the form

$$\eta\left(x; \theta_k^\eta\right) = {\theta_k^\eta}^\top x.$$

But it should be observed that this linearity, i.e. the linearity with respect to the states $x \in \mathscr{X}$, is not necessary for identification purposes. Here, only the linearity in the parameters is decisive. Often, one considers affine-linear models with an additional constant:

$$\eta\left(x; \theta_k^\eta\right) = {\theta_k^\eta}^\top x + \theta_{k0}.$$

One could also use polynomials of a higher degree. For example, in dimension $d = 2$, we have the second order polynomials

$$\eta\left(x; \theta_k^\eta\right) = a_k x_1^2 + b_k x_1 x_2 + c_k x_2^2 + d_k x_1 + e_k x_2 + f_k$$

with $\theta_k^\eta := (a_k, b_k, c_k, d_k, e_k, f_k) \in \mathbb{R}^6$.

Normalized radial basis functions We return to the parameterized weight functions $w_k(x) := w\big(x;\theta_k^w\big)$ for $k = 1,\ldots,N$. We have to fulfill the normalizing conditions mentioned above. Given any set of parameterized functions $\tilde{w}(x;\theta_k^w)$, $k = 1,\ldots,N$, we are able to force these functions to fulfill the normalizing conditions by just normalizing them. Doing this, the ***normalized*** weight functions are

$$w\big(x;\theta_k^w\big) := \frac{\tilde{w}(x;\theta_k^w)}{\sum_{j=1}^N \tilde{w}(x;\theta_j^w)}.$$

A possible choice for $\tilde{w}$ are the ***radial basis functions*** (Gauss bells)

$$\tilde{w}(x;\theta_k^w) := \exp\Big(-\frac{1}{2}\big(h^w(x)-\mu_k\big)^\top \Sigma_k \big(h^w(x)-\mu_k\big)\Big)$$

where $h^w : \mathscr{X} \longrightarrow \mathbb{R}^m$ denotes a known function. For some state $x \in \mathscr{X}$ the image $h^w(x)$ is called ***regime vector***. In the simplest case, all Σ_k are diagonal, and the parameters are

$$\theta_k^w := (\mu_k,\sigma_k) \in \mathbb{R}^m \times (\mathbb{R}^+)^m, \qquad \Sigma_k := \operatorname{diag}(1/\sigma_{k,1}^2,\ldots,1/\sigma_{k,m}^2).$$

Decision trees and regimes We have two kinds of parameters: the ***location parameters*** μ_k and the ***scale parameters*** σ_k. Both can be computed from a given rectangular partition of the regime space $\Psi := h^w(\mathscr{X})$. If the regime space Ψ is a hypercuboid, such a partition is given by a disjoint union of subsets of Ψ which are by itself hypercuboids. The partition is described easily by a decision tree (a detailled definition will follow in subsection 1.4.1). Both, an example of a partition for a 2-dimensional regime space Ψ and the corresponding decision tree are shown in figure 1.5.

The parameters μ_k and σ_k are then chosen as follows:

- μ_k: as the middle point of the k-th regime,
- σ_k: proportional to the edge lengths of the k-th regime.

An example of the resulting global model given by a local model network is shown in figure 1.6, together with the partial models. In this figure, the partial models are restricted to their regimes, but actually they are valid over the whole regime space. The localization is provided only by the weight functions (the weight functions are not shown in the figure).

Remark: We have derived local model networks as a generalization of the basis function approach such that the weight functions correspond to the linear parameters α_k of the basis function decomposition

$$\sum_k \alpha_k \eta_k(x).$$

The partial models Γ_k correspond then to the basis functions η_k. With the same right we could have done it in the opposite way: the weight functions are the basis functions, and the parameters α_k are generalized to the partial models Γ_k. This ambiguity in interpretation may be a hint that the separation into weight functions and partial models is not a natural but only a pragmatical one: it separates the nonlinear from the linear parameters. In chapter 5 we will reunite both weight functions and partial models into basis functions given by a wavelet decomposition.

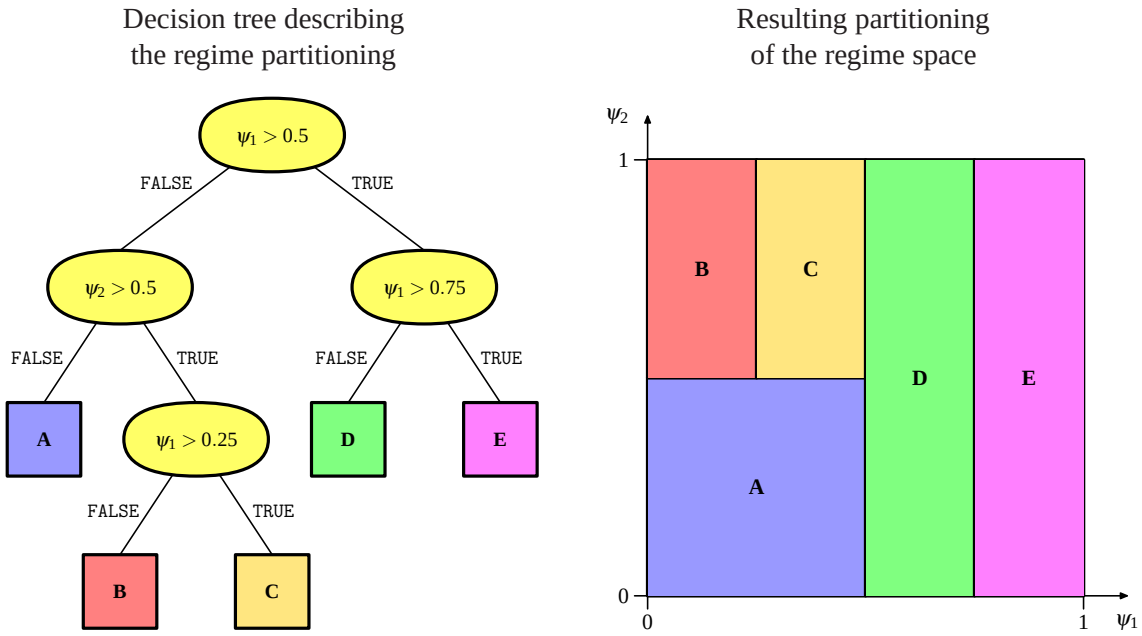


Figure 1.5: Decision tree and corresponding partition of the regime space Ψ

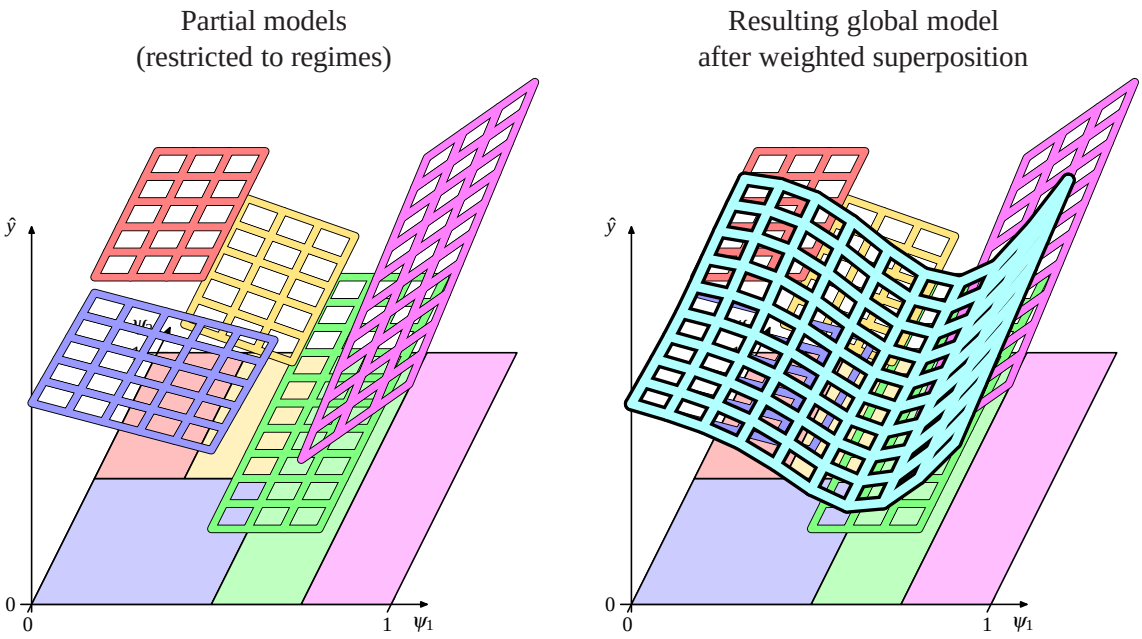


Figure 1.6: Partial models and global model

1.3 The LOLIMOT algorithm

Given input/output data of a real system, how can we identify/construct a local model network fitting to these data? The ***LOLIMOT algorithm*** serves to this purpose. The term LOLIMOT stands for **LO**cal **LI**near **MO**del **T**ree. The algorithm was developed by Oliver Nelles (Nelles [1997], see also Nelles [2001]).

The algorithm The idea behind the algorithm is simple: beginning with an overall identified global linear model the algorithm divides this model into two local linear models, each of them reigning on its own regime. They are identified separately on their respective regime and superposed in a smooth way. In the next step, out of these models the one is chosen which fits worst to the data. This model is further divided into two new local models. This procedure will be repeated iteratively until a certain stopping criterion is fulfilled (see figure 1.7 on the left). The algorithm is greedy in the sense that in each step it picks out the best choice available during this step. It is nevertheless not guaranteed that this choice is also the best choice concerning the overall model because a decision once made will never be revised. The resulting model is thus not optimal. The algorithm works only from coarse to fine, a step backwards is not possible. If one draws a picture of the generations of models created during a run of the algorithm, one recognizes a growing tree-like structure. On the root we will find the first global linear model, its descendant branches designate the models produced in the first division step and so going on onto the leaves where the active local models are located. A similar picture could be drawn concerning the underlying regime spaces where the local models live on. Then the resulting tree will have the same structure but its branching points will be labelled with the regimes resulting from the partitioning of the regime space (see the right part of figure 1.7). If we put together all regimes on the leaves we get a disjoint union of the complete regime space.

Choice of local model and split The critical points of the algorithm are the decisions it has to make: the choice of the local model to split, and the decision how the split should be done. The choice of the model is based on an ***error function*** which measures the error between the output of a considered local model and the measurements observed on the real system. Basis for the error function $e(\Gamma)$ of the global model Γ is the simulated NOE-output $\hat{y}(t)$:

$$e^2(\Gamma) := \|y(\cdot) - \hat{y}(\cdot)\|_2^2 = \sum_t |y(t) - \hat{y}(t)|^2.$$

The error function $e_k(\Gamma_k)$ for one local model Γ_k for some k is then obtained via a certain weighting of e:

$$e_k^2(\Gamma_k) := \sum_t w_k(x(t))|y(t) - \hat{y}(t)|^2.$$

In this way an error for each local model is obtained and the model with largest error will be chosen. The second decision to make is much more involved: how to split optimally the chosen model into two new models? This actually comes back to decide how the regime of the chosen model is to be partitioned into two disjoint regimes. Of course, without any restrictions on the

possible splits this is a far too difficult problem. In the original algorithm, the regime space is always a hypercuboid in the vector space $\mathbb{R}^n$. Only splits which are given by an axis-parallel hyperplane are allowed. Splitting the original hypercuboid in the first step of the algorithm results thus in two new hypercuboids. These hypercuboids can be divided in the same way. We see that the regimes are always hypercuboids. But the number of possible splits, say: the number of possible hyperplanes intersecting a given hypercuboid, is still too high. Therefore the original algorithm does allow only splits which divides the hypercuboid into two equal-sized halves. This procedure is called ***dyadic partitioning***. The number of possible splits is thus drastically reduced: If the regime space is lying in $\mathbb{R}^n$, only the choice between n splits remains. The algorithm just tests all these splits, identifies the local models, computes the error function for all these models, and chooses the split which yields the smallest overall error. Apart from splitting the hypercuboids only once into two new hypercuboids, also $k-1$ splits into k equal-sized hypercuboids are possible (Nelles [1997]).

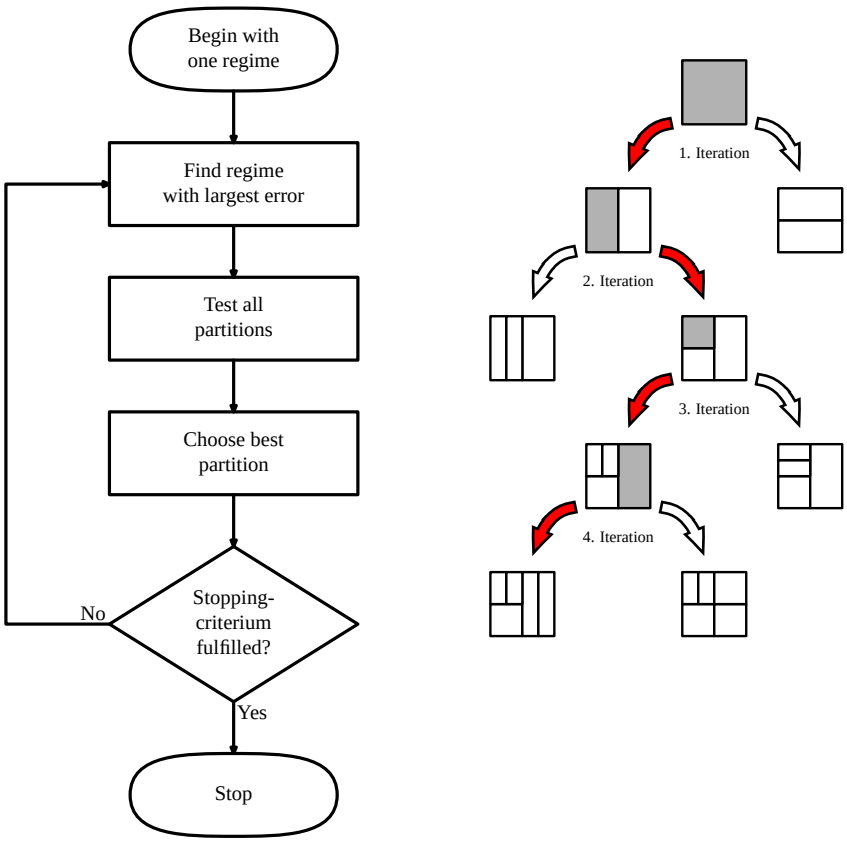


Figure 1.7: The LOLIMOT algorithm (left); example of a growing tree structure and partitioning of regime space (right)

Identification of local ARX models

Another question is how to identify the local models. This question is of course strongly related to the question of which model type is used for the local models. In the original

algorithm linear ARX models constitute these local models. They have the advantage that they are not only linear in the input of the model, but also, and this is much more important, in the parameters which have to be identified. So, simple linear regression can be used for identification. Of course, when combining the local models to a global model, one has to modify the identification algorithm slightly. Nelles [1997] proposes two types of parameter estimation procedures (see also Nelles [2001]): global estimation and local estimation. Before going into details we fix some notational conventions. Given a local model network

$$(\Gamma u)(t) = \sum_{k=1}^{N} w_k(x(t))\eta_k(x(t))$$

with local ARX models, we may write

$$\eta_k(x(t)) = \theta_k^\eta h^\eta(x(t))$$

with $d \times n$-matrix θ_k^η and $n \times 1$ regression vector $h^\eta(x(t))$ (d being the dimension of the output space $\mathscr{Y}$ and n being the dimension of the regression space $\Psi := h^\eta(\mathscr{X})$). The estimation can be done for each component of the output $\hat{y}$ separately. The j-th component corresponds to the j-th rows of the matrices θ_k^η. For notational convenience we therefore assume in the following that $d = 1$. The matrices θ_k^η are then $1 \times n$ row vectors. For general d, the procedures have to be repeated d times.

Global estimation For the global estimation procedure one needs to recognize that the parameters θ_k^η remain linear in the global model:

$$(\Gamma u)(t) = \sum_{k=1}^{N} w_k(x(t))\theta_k^\eta h^\eta(x(t)) = \sum_{k=1}^{N} \theta_k^\eta \left(w_k(x(t))h^\eta(x(t))\right) = D(t)\theta^{\eta\top}$$

with the $1 \times Nn$ row vector

$$\theta^\eta := (\theta_1^\eta, \ldots, \theta_N^\eta)$$

and the $1 \times Nn$ row vector

$$D(t) := \left(w_1(x(t))h^\eta(x(t))^\top, \ldots, w_N(x(t))h^\eta(x(t))^\top\right).$$

The global estimation procedure estimates in each iteration the complete parameter vector $\theta^\eta = (\theta_1^\eta, \ldots, \theta_N^\eta)$ by the least squares method: we have to find the parameter vector θ^η which minimizes the error function

$$e(\Gamma) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2$$

where we assume that we are given M measurements $y(t_j)$ and corresponding model outputs $\hat{y}(t_j)$ and set

$$\mathbf{y} := y(t_j)_{j=1}^M \qquad \text{and} \qquad \hat{\mathbf{y}} := \hat{y}(t_j)_{j=1}^M.$$

The number of measurements M has to be greater than the number of scalar parameters Nn. The model outputs $\hat{y}(t_j)$ lead to a system of equations

$$\hat{\mathbf{y}} = \mathbf{D}\theta^{\eta\top}$$

with the $M \times Nn$-matrix

$$\mathbf{D} := \begin{pmatrix} D(t_1) \\ D(t_2) \\ \vdots \\ D(t_M) \end{pmatrix}.$$

The optimal parameter $\hat{\theta}^{\eta}$ minimizing the error function can be attained by the solution of the so-called normal equations:

$$\hat{\theta}^{\eta\top} = (\mathbf{D}^\top\mathbf{D})^{-1}\mathbf{D}^\top\mathbf{y}.$$

If the matrix $\mathbf{D}$ is ill-conditioned, the inversion of $(\mathbf{D}^\top\mathbf{D})^{-1}$ leads to numerical instabilities and one should use other methods, e.g. using the pseudo-inverse $\mathbf{D}^\dagger$ (computed by means of the singular value decomposition of $\mathbf{D}$) and setting

$$\hat{\theta}^{\eta\top} = \mathbf{D}^\dagger\mathbf{y}.$$

The computational complexity of the inversion of the matrix $(\mathbf{D}^\top\mathbf{D})$ is

$$O((Nn)^3),$$

the computation of the pseudo-inverse $\mathbf{D}^\dagger$ is even more involved. For general $d \geq 1$ we have thus at least the complexity

$$O(d(Nn)^3).$$

Local estimation The local estimation procedure uses the fact that in each LOLIMOT iteration the parameters change only locally: a split at a vertex u produces two new models Γ_{u_1} and Γ_{u_2} at the children u_1 and u_2 of u. The parameters of the corresponding output functions η_{k_1} and η_{k_2} have to be identified, the parameters of the other local models remain unchanged. If the weight functions were disjunct (not overlapping), this procedure would be equivalent to global estimation. In the case of overlapping weight functions (as the normalized Gaussian bell functions are), an error is introduced which is assumed to be negligible. Local estimation is done by weighted least squares, the weights being given by the weight functions: for a given local model

$$(\Gamma_k u)(t) = w_k(x(t))\theta_k^{\eta} h^{\eta}(x(t)) = \theta_k^{\eta}\left(w_k(x(t))h^{\eta}(x(t))\right) = D_k(t)\theta_k^{\eta\top}$$

with the $1 \times n$ row vector

$$D_k(t) := w_k(x(t))h^{\eta}(x(t))^\top,$$

we want to find the $1 \times n$ row vector θ_k^{η} which minimizes the error function

$$\|\mathbf{y} - \hat{\mathbf{y}}\|_{2,\mathbf{w}_k}$$

where $\|\cdot\|_{2,\mathbf{W}_k}$ denotes the weighted square norm

$$\|z\|_{2,\mathbf{W}_k} := \sqrt{z^\top \mathbf{W}_k z} \qquad \text{for } z \in \mathbb{R}^M,$$

with the $M \times M$ diagonal matrix

$$\mathbf{W}_k := \operatorname{diag}\big(w_k(x(t_1)), \ldots, w_k(x(t_M))\big).$$

Now M has only to be greater than n. Defining the $M \times n$-matrix

$$\mathbf{D}_k := \begin{pmatrix} D_k(t_1) \\ D_k(t_2) \\ \vdots \\ D_k(t_M) \end{pmatrix},$$

the weighted least squares estimator is given by

$$\hat{\theta}_k^{\eta\top} = (\mathbf{D}_k^\top \mathbf{W}_k \mathbf{D}_k)^{-1} \mathbf{D}_k^\top \mathbf{W}_k \mathbf{y}.$$

Also here, the matrix we have to invert may be ill-conditioned, and we could e.g. use the pseudo-inverse $(\mathbf{W}_k\mathbf{D}_k)^\dagger$ leading to

$$\hat{\theta}_k^{\eta\top} = (\mathbf{W}_k \mathbf{D}_k)^\dagger \mathbf{W}_k \mathbf{y}.$$

The computational complexity for the matrix inversions (not using the pseudo-inverse) involved in estimating M local models is now only

$$O(Mn^3),$$

and for general $d \geq 1$ we have thus a complexity of

$$O(dMn^3).$$

One has gained a factor of M^2: the complexity of the LOLIMOT algorithm with local estimation grows only linearly with the number of local models M. Similar arguments hold when using the pseudo-inverse.

Modelling of uncertainties It should be noted that, in the LOLIMOT algorithm, the uncertainties like measurement or process noise are only modelled implicitly. The LOLIMOT algorithm separates the estimation problem into two steps, the structure and the parameter optimization. The structure optimization is given by the constructive determination of the (nonlinear) weight parameters, whereas the parameter optimization concerns the regression step for the (linear) parameters of the local models. As already Nelles [1997] notes, for the structure optimization the NOE-error is used, whereas for the parameter optimization the ARX-error is used. In this sense, the overall error is hybrid, and even assuming Gaussian errors in the outputs, the estimation is biased. Nelles [1997] points out the possibility of a later iterative NOE-optimization with iterative (e.g. gradient-based) algorithms.

Stopping criterion The algorithm will stop if some stopping criterion is fulfilled. This could be after a fixed number of steps or better by applying some statistical or information theoretical model selection criterion.

Model selection Model selection is of general interest, and we shortly review some of the most common model selection criteria. ***Model selection*** usually aims at an optimal choice concerning the trade-off of goodness-of-fit versus model complexity. This is a bias/variance trade-off: Inference under models with too few parameter can be biased, while models with too many parameters may lead to identifications of effects which actually belong to the noise (Burnham and Anderson [2004]). Model selection should therefore be seen as a regularization technique. This trade-off is obtained by adding a suitable penalizing term to the regression error and by selecting the model and parameters that minimize this augmented term. A special case of model selection is variable-selection in regression. In the following, we assume that n data $y_1,\dots,y_n$ are observed, and one wants to choose a model out of a set $\mathscr{M}$ of possible models, indexed by parameter vectors $\theta \in \Theta$. In this setting, models and parameters are in a one-to-one relation. A two-step procedure is necessary: First, one has to choose the model structure (given by the dimension p of the parameter vector θ; selections step) and then, one has to estimate the values of the parameters (estimation step; see Foster and George [1994]). In some situations, e.g. in wavelet regression (see chapter 4), there is a maximal value m for the dimension of the parameters (e.g. wavelet coefficients), which plays a rôle for some information criteria.

The model selection criteria reviewed here are all based on maximum likelihood estimation (for alternative Bayesian approaches see chapter 3). By $\ell(\theta;y)$ we denote the likelihood of the parameter θ given the data y which is to be maximized. If we assume Gaussian innovations with unknown variance σ^2, then the log-likelihood can be given in terms of the Mean Square Error, because in this case

$$\log \ell(\theta,\sigma^2;y) = -\frac{1}{2}\sum_{i=1}^{n}\frac{\varepsilon_i^2(\theta)}{\sigma^2} - \frac{n}{2}\log\sigma^2 - \frac{n}{2}\log 2\pi$$

with

$$\varepsilon_i(\theta) := \mathbf{y}_i - \hat{\mathbf{y}}_i(\theta)$$

(see e.g. Ljung [1999]). Thus, if σ^2 is assumed to be known, minimizing $\log\ell(\theta,\sigma^2;y)$ essentially results in minimizing

$$\varepsilon_i^2 = \|\mathbf{y}-\hat{\mathbf{y}}\|_2^2 = \mathbf{e}^2.$$

Maximum likelihood estimation leads to choosing the highest possible dimension. Therefore, the model selection criteria based on maximum likelihood usually add a term which penalizes the model complexity, e.g. the dimension of the parameter vector.

The earliest model selection criterium is the ***Akaike Information Criterion, AIC***, Akaike [1973]. It is given by maximizing the term

$$\log\ell(\theta;y) - p$$

and originated as a method to minimize the expected Kullback-Leibler distance of the fitted model to the true model (see e.g. Foster and Stine [1997]). The AIC criterium is known to

overfit when presented with data originating from a finite-dimensional model: it yields models with too many parameters. The ***Bayesian Information Criterion, BIC***, also called ***Schwarz's Information Criterion, SIC*** (Schwarz [1978]) has a larger penalty term compared to the AIC, and avoids this overfitting. The BIC/SIC is given by maximizing

$$\log \ell(\theta; y) - \frac{p}{2} \log n.$$

The resulting models are thus more parsimonious than those obtained with AIC. The BIC was derived by Schwarz [1978] as a limiting case of Bayesian estimators, hence the name; but it should be noted that the criterion itself is not Bayesian (it is not depending on any prior distributions, cf. chapter 3).

Among the model selection criteria which assume a maximal number m of parameter dimensions is the ***Risk Inflation Criterion, RIC***, of Foster and George [1994]. It is obtained by maximizing

$$\log \ell(\theta; y) - p \log m$$

and is based on considerations of the risk inflations, i.e. the maximum increase in risk due to selecting rather than knowing the „correct" model. The same bound $p \log m$ was obtained by Donoho and Johnstone [1994] for hard thresholding in the context of wavelets, see chapter 4.

There are many other model selection criteria, and there is no best one: Different classes of problems yield different optimal selection criteria. There are several trials for unification of these different criteria, e.g. Foster and Stine [1997] use an information-theoretical approach where a model is seen as a reduction (compression) of the observed data, and model selection is thus the task to look for the best compression in terms of the number of bits which is needed to describe the models (parameters) selected and the associated values. The different model selection criteria then result from different model representations.

1.4 Problems and possible improvements

For some nonlinear systems, the LOLIMOT identification algorithm has shown to work well. But nevertheless, problems with this identification scheme arise concerning:

- the identifiability of models,
- the estimation of the parameters,
- the use of prior knowledge,
- the convergence of the global model,
- the extrapolation behaviour,
- the model size (number of parameters).

In the following sections we will discuss each of these points separately and provide some possible solutions. While in this chapter we sketch some immediate improvements which alter the original algorithm only slightly, we will present basic theories which lead to an alternative algorithm in the subsequent chapters. This new algorithm will be able to identify a wider range of nonlinear systems with a better justification of its usefulness regarding nonlinearity and non-normality (non-Gaussian noises). The price to be paid is that it needs more resources for identification, like computation time and computer memory. But once identified, the resulting model is much more compact and so faster during simulation than the original one. We will come back to all this in the chapters 2 to 5. For the time being, we discuss shortly problems and possible improvements of the original LOLIMOT algorithm, and discuss afterwards in more detail the easier ones of them.

Limited types of models

Problem The original LOLIMOT algorithm uses only ARX models derived from deterministic difference equations. These models cannot handle long memory effects (like hysteresis), as will be explained in chapter 2. The ARX models look back into the past only a fixed number of time steps. This fixed number is given by the values n_u and n_y. The modelling of long memory effects like hysteresis needs typically a memory which is not restricted in size. Another point already mentioned is that ARX models are not well suited for simulation purposes because they are based on measured outputs of the real system which are often not available.

Possible solutions A solution is to provide a wider range of basic models and combinations of them, e.g. for hysteresis, and to include OE models for simulation purposes (based on calculated output).

As an example of the incorporation of a long memory effect let us mention the Preisach model as basis model. The Preisach model requires a so-called representation string $r(t)$ which has to be updated at each time step t. A memory element for the storage of this string could replace the tapped delay structure used for difference dynamics in the original local model network, see figure 1.8 and compare to figure 1.3 on page 9. The state $x(t)$ is then given by this representation string $r(t)$. This idea will be developed in more detail in chapter 2.

Least squares estimation not suited for nonlinear identification

Problem The original algorithm uses a linear weighted least squares (LS) algorithm for parameter identification. This is equivalent to Maximum Likelihood (ML) estimation if the parameters are linear and model as well as data errors are Gaussian. This estimation provides also confidence intervals for the estimated parameters. If the parameters are *not* linear or the data error is *not* Gaussian, then the least squares estimation is only approximately valid. There is no possibility to say how good this approximation is in a concrete application. For strong nonlinearities, parameter estimation and confidence intervals can be completely wrong. We will go into more details concerning the estimation problem in chapter 3.

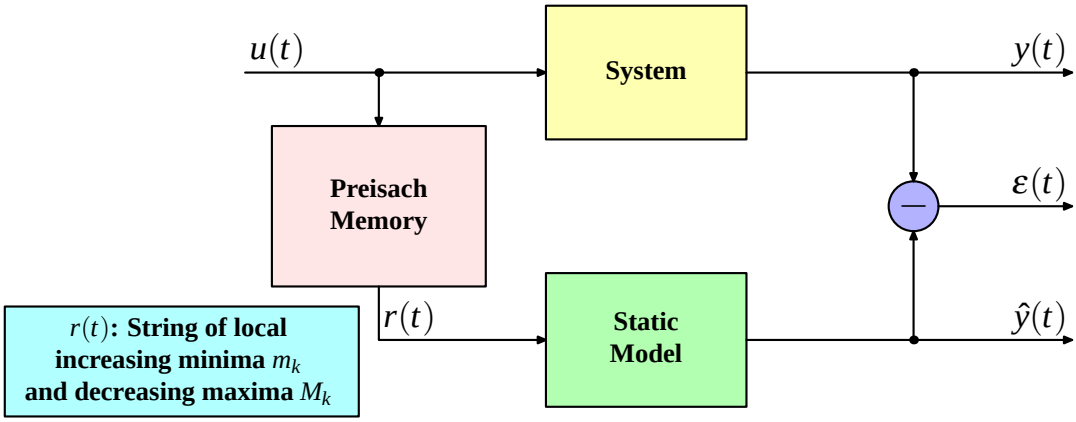


Figure 1.8: Separation of dynamics with Preisach memory

Possible solutions To solve this problem, use stochastic models like stochastic state space models or more general graphical models, also called belief networks, and replace the least squares estimation by an estimation more suited for this kind of models, e.g. Bayesian estimation in combination with decision theory. We give a sketchy idea where the difference lies between deterministic and stochastic models in table 1.1.

	Deterministic Models	Stochastic Models	Knowledge Used
Given by	Operators on Hilbert spaces	Stochastic Pocesses	
Data Error	Bounded	White Noise	Data (Black Box)
Estimation	Inversion + Regularization (“Ill-Posed Problem”)	Bayes Inference + Loss function (“Unstable Problem”)	
A priori knowledge	Regularity	Prior Distributions	Structural (White Box)

Table 1.1: Deterministic versus stochastic models

Very restricted usage of prior knowledge

Problem Prior knowledge is used only to decide the model type. No smoothness and regularity properties of the real system or signals are used. The local model network and its identification algorithm provide a black-box model which is quite black. The only grey ingredient is the interpretability of the local models. But notice that these local models are actually also black-box models. There is also no way to put constraints on the parameters.

Possible solutions A solution may be to use regularization techniques in the deterministic case or Bayesian techniques where prior knowledge is provided by prior distributions. Bayesian statistics may be used as "stochastic regularization". Bayesian probability theory is treated in chapter 3.

A short word concerning the notion of regularization: As every identification or estimation process must be seen as an inverse problem, i.e. as a problem where the result is given and the cause is sought, one has to be careful because slight errors in the given results can lead to arbitrarily large errors in the estimated causes ("inverse and ill-posed problems"). Instable algorithms arise if no care is taken. To circumvent these instabilities, regularization methods must be applied. These methods consist usually in approximating the original problem by a family of slightly modified but stable problems. To do this, always prior knowledge about the sought solution is needed. In deterministic settings, this prior knowledge is given by smoothness conditions on the solution like differentiability. In Bayesian settings, the prior knowledge is always given by prior distributions for the sought values.

Convergence and convergence rates

Problem Very few is known about the convergence properties of artificial neural networks with activation functions of sigmoidal or radial basis shape. As the local model network can be interpreted as a special kind of neural network, the same is true in this case.

Possible solutions Our solution will be to use wavelets and multiresolution analysis. For these issues, an elaborated mathematical theory exists, and wavelets are successfully used for nonlinear approximation of functions. Approximation theory offers here good convergence results, and the functions which can be well approximated are characterized by well-understood function spaces, the Besov spaces. These in turn can be characterized by sparsity properties of wavelet coefficients. The close interconnections between these three different theories, wavelet theory, approximation theory, and functional analysis, will be treated in chapter 4.

How can function approximation help for system identification? The answer is given by the separation of the dynamics. In the case of difference basis models like ARX models we only have to identify the static functions $\eta_k : \mathscr{X} \longrightarrow \mathscr{Y}$. In the above mentioned case of Preisach hysteresis basis models, we will show in chapter 2 that a kind of primitive function can be derived which completely describes the Preisach model. In both cases, the identification of the model is reduced to the identification of a function

$$f : \mathscr{D} \longrightarrow \mathbb{R}^n \qquad \text{with } \mathscr{D} \subseteq \mathbb{R}^d$$

or, better said, the approximation of this function. This in turn can be done by the established methods of Constructive Approximation for functions, developed in Approximation Theory. The main idea for the identification of f is:

- For all $m \in \mathbb{N}$, choose nested sets X_m of functions mapping $\mathscr{D}$ to $\mathbb{R}^n$.
- Choose $m \in \mathbb{N}$ and a function $g \in X_m$ which approximates f best (in a certain sense).

Extrapolation behaviour of model is extremely bad

Problem The normalization procedure of weight functions leads to unexpected effects in the extrapolation area, that is the area of the regime space outside the hypercuboid which is used for identification. Even regimes inside the interpolation area have distorted boundaries. More details will be shown in a subsequent section.

Possible solutions Since the problem lies in the normalization procedure, one should try to avoid it and to use weight functions which are already normalized by themselves.

A possibility to produce readily normalized weight functions is easily derived when we translate the LOLIMOT decision tree into fuzzy rules. We will follow an idea of Eva Barrena (see [Barrena Algara, 2007]). The decision tree can be translated step by step into corresponding fuzzy rules. Interpreting the binary operator $>$ in an adequate way, for example as fuzzy operator, we get normalized smooth weight functions. More in a subsequent section of the present chapter.

Models are too large

Problem Due to the restrictions of the subdivision algorithm (axis-parallel and dyadic subdivisions) the size of the global model can be very large. By a large model size we mean a model with many local models and thus many parameters. This is not favourable because it reduces interpretability and increases computation time during simulation. The algorithm does not work on scales, and so the parameters do not decay if model size increases, so these parameters cannot be neglected (no thresholding is possible). One reason for the large model sizes is that the algorithm works only from “coarse to fine”.

Possible solutions As solution, again the use of wavelets may help: Multiresolution analysis works on scales. Another possibility is the development of pruning methods: “from fine to coarse”, as well as the development of more flexible subdivisions: not axis-parallel and/or non-dyadic subdivisions. More details will be given in some subsequent section in this chapter.

Summary of improvements of the LOLIMOT algorithm

We summarize the possible improvements:

- More flexible basic models (hysteresis, simulation),
- Stochastic models,
- Bayesian estimation (“stochastic regularization”),
- Normalized weight functions and/or wavelets,
- Mulitresolution analysis,
- Nonlinear approximation,

- More flexible decision trees (subdivisions),
- Pruning (coarse to fine *and* fine to coarse).

The immediate improvements of the LOLIMOT algorithm in the subsequent sections will be:

- We want to allow more splits, not only dividing the cuboids into two equal halves, but providing other split ratios or even diagonal splits.
- We will add a gradient based optimization algorithm to optimize the parameters of the whole model, especially
  - the parameters belonging to the weight functions,
  - the parameters belonging to the local models, especially after changing the model type from ARX to OE.
- We will provide a pruning mechanism which cuts away unnecessary branches of the tree.

All these modifications of the original LOLIMOT algorithm will of course yield a longer computation time. But they will also establish smaller models, i.e. the global model will need less partial models than with the original algorithm to perform the same accuracy. Anyhow, we did not implement the proposed improvements because even with these improvements the LOLIMOT algorithm is not able to identify systems with long-time memory like hysteresis.

1.4.1 Decision trees und weight functions

Our aim now is to define normalized weight functions using directly the structure of decision trees. We first provide the necessary graph-theoretical foundations, eventually turning to a definition of decision trees suited to our purposes. It is then easy to construct normalized and flexible weight functions.

Finite graphs

In this section we provide the basic notions of graph theory which are needed here. The graph terminology differs more or less among different authors or for different usages. We follow roughly Lauritzen in Barndorff-Nielsen et al. [2001] for the graph terminology.

Definition of graphs We begin with the definition of graphs.

Definition 1.1 (Graphs)**:**

- *We call a pair $G = (V,E)$ a **(finite) graph**, if V is a finite set and $E \subseteq V \times V$ is a binary relation of V. We call V the **vertices** (or **nodes**), and E the **edges** (or **links**) of the graph G. Given an edge $(u,v) \in E$, the vertices u and v are called the **endvertices** of this edge. The graph G is called **complete** if $E = V \times V$.*

- *We call an edge $(u,v) \in E$ **undirected** if the reversed relation (v,u) is also in E. We write then $u \sim v$ and call u and v **neighbours** of each other. We denote the set of all neighbours of a vertex $u \in V$ by* $\mathrm{ne}(u)$.
- *We call an edge $(u,v) \in E$ **directed** if $(v,u) \in E$ is not in E. We write $u \to v$. In this case, u is called the **parent** of v, and v is called the **child** of u. We denote the set of all parents of a node $u \in V$ by* $\mathrm{pa}(u)$ *and the set of all children of u by* $\mathrm{ch}(u)$.
- *We call the graph G **undirected** if all edges are undirected, i.e. if the relation E is symmetric. We call the graph G **directed** if all edges are directed.*
- *We call an edge $(u,u) \in E$ for some $u \in V$ a **loop**. A Graph without loops is called a **simple graph**.*

Graphs are often pictured as points representing the vertices V which are connected by lines and arrows according to the relations given by the edges in E. For a directed edge $(u,v) \in E$, an arrow is drawn with the head pointing in direction of v. If an edge $(u,v) \in E$ is undirected, then this edge and its reversed edge $(v,u) \in E$ are pictured as only one line connecting the points u and v, without arrow head (see figure 1.9).

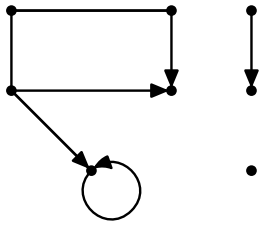


Figure 1.9: A graph with 8 vertices and directed and undirected edges

Definition 1.2 (Subgraphs and Cliques)**:** *Let $G = (V,E)$ be a graph. We call $G_A = (A,E_A)$ a **subgraph** of G if $A \subseteq V$ and $E_A \subseteq E \cap (A \times A)$. If $E_A = E \cap (A \times A)$ then we call G_A the subgraph of G **induced** by A. A maximal complete subgraph is called **clique** of G.*

The subgraph G_A may contain the same vertex set as the graph G, i.e. $V = A$, but possibly fewer edges. Cliques will occur in chapter 3 in connection with graphical models.

Degree Counting the ingoing and outgoing edges of a given vertex leads to the notion of the degree of a vertex.

Definition 1.3 (Degree of a vertex)**:** *Let $G = (V,E)$ be a graph, and let $u \in V$ be a vertex. We define the **outdegree** d_{out} of u to be the number of its children,*

$$d_{\mathrm{out}} := \#\mathrm{ch}(u) = \#\{v \mid u \to v\},$$

*the **indegree** d_{in} of u to be the number of its parents,*

$$d_{\mathrm{in}} := \#\mathrm{pa}(u) = \#\{v \mid v \to u\},$$

*and the **degree** d of u to be the number of its neighbours,*

$$d := \#\mathrm{ne}(u) = \#\{v \mid u \sim v\}.$$

(We denote the number of elements in a finite set A by $\#A$.)

Paths in a graph We can follow several edges inside a graph. We get paths and cycles. The definitions differ slightly from author to author; we will fix them as follows:

Definition 1.4 (Paths, cycles, trails)**:** *Let $G = (V,E)$ be a graph. A sequence $(u_0,\ldots,u_n)$ of pairwise distinct vertices $u_0,\ldots,u_n \in V$ with $n \in \mathbb{N}$ is called **path** of length n if all binary relations (u_i,u_{i+1}), $i = 1,\ldots,n-1$, are in E. A **cycle** is a path $(u_0,\ldots,u_n)$ with $u_0 = u_n$. In contrast, a sequence $(u_0,\ldots,u_n)$ of pairwise distinct vertices $u_0,\ldots,u_n \in V$ with $n \in \mathbb{N}$ is called **trail** of length n if for all $i = 1,\ldots,n-1$ either $(u_i \to u_{i+1})$, $(u_{i+1} \to u_i)$ or $(u_i \sim u_{i+1})$. The graph G is called **connected** if for each $u,v \in G$ there exists a trail $(u_0,\ldots,u_n)$ with $u_0 = u$ and $u_n = v$. A maximal connected subgraph induced by G will be called a **connected component** of G.*

Paths and cycles thus follow always the direction of the edges and do not cross itself. A trail may go against the direction, but still does not cross itself. The graph in figure 1.9 has 3 connected components.

Acyclic graphs: DAGs and trees Graphs without cycles are important in many applications. In chapter 3 we will consider graphical models, i.e. statistical models based on an underlying graph. An important case is given by the following definition:

Definition 1.5 (Directed acyclic graph (DAG))**:** *A graph is called a **directed acyclic graph (DAG)** if all edges are directed and there are no (directed) cycles.*

Whereas DAGs exclusively have directed edges, forests and trees are undirected:

Definition 1.6 (Forests and trees)**:** *We call an* undirected *graph $G := (V,E)$ a **forest** if it has no cycles. We call G a **tree** if G has no cycles and is connected.*

The connected components of a forest are thus trees. There are many equivalent characterizations of trees which can be found in every book on graph theory. We will just mention:

Theorem 1.1: *Let $G = (V,E)$ be an undirected connected graph. The following conditions are equivalent:*

(a) G is a tree, i.e. G has no cycles.

(b) We have $\#V = \#E - 1$.

(c) For arbitrary vertices $u,v \in V$ there exists at most one path from u to v.

Rooted trees Until now we have no ordering on the set of vertices in our trees: trees are undirected graphs by definition, and thus an ordering cannot be derived by the direction of edges. To enforce an ordering on a given tree $G = (V,E)$, we only need to choose one vertex out of V which shall be on the top of the tree. We will call this vertex the root of the tree G.

Definition 1.7 (Rooted tree)**:** *We call the pair $T = (G,r)$ a **rooted tree** if $G = (V,E)$ is a tree and $r \in V$ is a vertex, called the **root** of G. The **canonical ordering** $\geq$ of (G,r) is then defined by*

- *$u \geq v$ for $u,v \in V$ if u lies on the path from r to v.*

For simplicity, we often suppress mentioning the root r and often call already G a rooted tree. It is easy to prove that the canonical ordering is indeed an ordering on the vertices V. As usual, one writes $u > v$ if $u \geq v$ but $u \neq v$, $u \leq v$ if $v \geq u$, and $u < v$ if $v > u$. The ordering $\geq$ has an additional property: it is directed (as ordering), i.e. for each two vertices $u,v \in V$, there is a vertex $w \in V$ such $w \geq u$ and $w \geq v$ (one simply may choose $w = r$).

With this ordering $\geq$, we may construct an associated *directed* graph $G^*(V,E^*)$ which is directed according to $\geq$: The vertices V are the same, and the directed edge $u \to v$ is in E^* if and only if $u \sim v$ and $u > v$. We will often switch between these two graphs without mentioning. With this interpretation, we may say: Given a vertex $u \in V$ of a tree $T = (G,r)$ with $G := (V,E)$, the vertices which lie directly above u are the ***parents*** of u,

$$\mathrm{pa}(u) = \big\{ v \in V \;\big|\; v > u \text{ and there is no vertex } w \in V \text{ with } v > w > u \big\},$$

and the vertices lying directly below u are the ***children*** of u,

$$\mathrm{ch}(u) = \big\{ v \in V \;\big|\; u > v \text{ and there is no vertex } w \in V \text{ with } u > w > v \big\}.$$

We can distinguish two kinds of vertices in a rooted tree:

Definition 1.8 (Inner vertices and leaves)**:** *Let $T = (G,r)$, $G = (V,E)$, be a rooted tree and let $u \in V$ be a vertex. We call u a **leaf** if $\mathrm{ch}(u) = \emptyset$, and **inner vertex** (or **inner node**) else.*

We usually denote the set of all leaves of a rooted tree T by $\mathscr{L}_T$, and the set of all inner nodes of T by $\mathscr{N}_T$. In this way, we get the disjoint union

$$V = \mathscr{N}_T \dot\cup \mathscr{L}_T.$$

In a rooted tree, there exists at most one parent for each vertex u. Otherwise, if we assumed that there are two different parents v_1 and v_2, then, since the ordering is directed, there would exist a third vertex v with $v \geq v_1$ and $v \geq v_2$. In this way we would have constructed a circuit, from u via v_1 to v and back via v_2 to u. But circuits are not allowed in trees, thus we have at most one parent. And most vertices have indeed a parent; the root r is the only exception. It has no parent by definition (because it is on the top of the ordering), and again since the ordering is directed, for each vertex u there exists a vertex v with $v \geq u$ and $v \geq r$. It follows $v = r$ from the last assertion, and thus $r \geq u$ from the first one, so u must either be equal to the root, or it has a parent (lying on the path from u to r).

Each vertex of a rooted tree can be assigned to a certain level:

Definition 1.9 (Height and levels of a rooted tree)**:** *Let $T = (G,r)$ be a rooted tree.*

(a) *We call the set $V_l \subseteq V$ the set of vertices of **level** l with $l \in \mathbb{N}$, if V_l contains all vertices $v \in V$ such that the path from r to v has length l.*

(b) *The **height** $h \in \mathbb{N}$ of T is the length of a maximal path in T beginning at the root r.*

Thus: The root builds the level 0, its children level 1, their children level 2, and so on. On the "highest" level h, the height of T, we find only leaves. But leaves can of course also appear on other levels.

k-ary trees As further enrichment of structure of rooted trees we want to introduce an enumeration of the neighbours of an arbitrary vertex $u \in V$.

Definition 1.10 (Enumerations)**:** *Let (G,r) be a rooted tree with $G = (V,E)$. An **enumeration** or **orientation** $q := (q_u)_{u \in V}$ of (G,r) is given by a family of maps*

$$q_u : \mathrm{ne}(u) \longrightarrow \mathbb{N} \cup \{-1\},$$

which assigns to each neighbour of u an integer such that

(a) *the number -1 is assigned to the parent v of u, i.e. $q_u(v) = -1$ if $\mathrm{pa}(u) = \{v\}$, and*

(b) *the map q_u is injective.*

*We call the enumeration **strict** if the image of each q_u is equal either to $\{-1,0,1,\ldots,d-2\}$ or to $\{0,1,\ldots,d-1\}$ where $d = \deg u$ is the degree of u.*

Given an enumeration, we can assign to each vertex in V a unique string $b(v) = (b_1,\ldots,b_l)$ with length $\#b = l$ equal to the level of v and $b_i \in \mathbb{N}$ for each $i = 1,\ldots,l$ in the following way: We take the path $(r = v_0, v_1, \ldots, v_{l-1}, v_l = v)$ from the root r to the vertix v, and set

$$b_i := q_{v_{i-1}}(v_i),$$

that is b_i is the number of v_i with respect to its parent v_{i-1}. We will call this the ***associated string*** of the vertex v and the enumeration q. (We will use the associated strings in chapter 4 in connection with wavelet packets.) The root has always $b(r) = ()$, the empty string.

Definition 1.11 (k-ary trees)**:** *Let $T = (G,r)$ be a rooted tree. We call T a (**full** or **proper**) k-**ary** tree if the root r has degree k and all other vertices have either degree $k+1$ or degree 1.*

Recall that the leaves are the vertices with degree 1. Thus all inner vertices of a full k-ary tree except the root have degree $k+1$. There is also the name k-***regular*** tree for this kind of tree used in the literature, but we want to avoid it because a *regular graph* is a graph where *each* vertex has degree k, and so the definitions are not congruent. We will often drop the word full (or proper) in the above definition, but it should be mentioned that in other contexts the meaning of k-ary could be that each vertex has *at most* degree $k+1$. The most prominent k-ary tree is the ***binary*** tree, where $k = 2$ (see figure 1.10).

For a full k-ary tree $T = (G,r)$ with $G = (V,E)$ and enumeration q, we usually assume that this enumeration is strict. Considering a binary tree and a node $u \in V$, we use the notion ***left child*** for the child v with $q_u(v) = 0$ and ***right child*** for the child v with $q_u(v) = 1$.

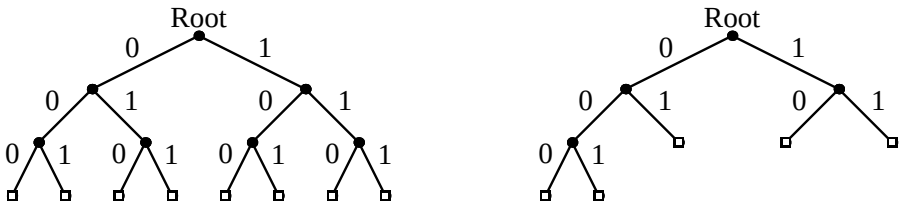


Figure 1.10: Binary trees with strict enumeration (the leaves are pictured as squares; the numbers -1 from the vertices to the parents are suppressed; the associated string for each node can directly be read by moving along the path from the root towards the node)

Probability trees As a last ingredient we want to add a weight to the edges.

Definition 1.12 (Edge weights and probability trees)**:** *Let $G = (V,E)$ be a tree. An **edge weight** w on G is just a non-negative map $w : E \longrightarrow \mathbb{R}_{\geq 0}$ such that for the undirected edges $(u,v) \in E$ the following symmetry condition holds:*

- $w((u,v)) = w((v,u))$.

*We say the edge weight is **normalized** if the edge weights to the children of u sum up to 1, i.e.*

$$\sum_{v \in \mathrm{ch}(u)} w(u,v) = 1.$$

*A **probability tree** (T,w) is a rooted tree $T = (G,r)$ together with a normalized edge weight w.*

Decision trees We are now ready to define the notion of a decision tree in a precise way:

Definition 1.13 (Decision trees)**:** *Let $T := (G,r)$ with $G = (V,E)$ be a rooted tree, Ω a set, and $\delta : \Omega \times E \longrightarrow \mathbb{R}$ a map. Then T is called **decision tree** on Ω with **decision map** δ if the following condition holds:*

- *For each $\omega \in \Omega$, the map $\delta(\omega,\cdot)$ is a normalized edge weight on G.*

*The **associated family of weight functions** $(w_u)_{u \in V}$ with $w_u : \Omega \longrightarrow \mathbb{R}$ is recursively defined by*

- $w_r(\omega) = 1$ *for all $\omega \in \Omega$ (where r is the root),*
- $w_u(\omega) = w_v(\omega) \cdot \delta(\omega,(v,u))$ *if v denotes the parent of u.*

Thus, for each fixed $\omega \in \Omega$, T together with $\delta(\omega,\cdot)$ is a probability tree. We usually consider only ***full binary decision trees***. If we additionally fix a strict enumeration $(q_u)_{u \in V}$, then we are able to distinguish for each inner vertex u between a left and a right child, i.e. the child with number 0 and 1, respectively, say v_0 and v_1. We then have a left edge (u,v_0) and a right edge (u,v_1), as well as a left edge weight $\delta(\cdot,(u,v_0))$ and right edge weight $\delta(\cdot,(u,v_1))$. From the definition of decision trees we get

$$\delta(\cdot,(u,v_0)) + \delta(\cdot,(u,v_1)) \equiv 1,$$

such that it is enough to provide one of these maps (we use the right edge weight!), which equally well might be assigned to the vertex u, by defining

$$\delta_u(\cdot) := \delta(\cdot,(u,v_1)).$$

We call these δ_u's the ***components*** of the decision map δ. This is the way we have pictured the decision map in the forgoing sections (for example in figure 1.5 on page 18). It follows

$$\delta(\cdot,(u,v_1)) = \delta_u(\cdot) \qquad \text{and} \qquad \delta(\cdot,(u,v_0)) = 1-\delta_u(\cdot).$$

The components δ_u of the decision map are thus defined for all inner nodes $u \in \mathscr{N}_T$.

Decision trees and logic If the components of the decision map are boolean functions, i.e. the image consists of the two-element set $\{0,1\}$,

$$\delta_u : \Omega \longrightarrow \{0,1\} \qquad \text{for all } u \in \mathscr{N}_T,$$

then boolean logic tells us that the operator $1-\cdot$ appearing in the above equation for the left edge decision map is exactly the `NOT` operator of boolean logic. It is then clear why the left edge of an inner node u can be labelled by `FALSE` and the right edge labelled by `TRUE`. The associated weight functions of the decision tree represent then boolean functions $w_u : \Omega \longrightarrow \{0,1\}$. If we return to the general decision maps with images on the whole interval $[0,1]$, we could use fuzzy logic to interpret the decision tree logically. Here, the `NOT` operator is still defined by $1-\cdot$. The labels `FALSE` and `TRUE` still make sense.

Decision trees and weight functions We will now show a simple lemma concerning the associated weight functions. It says that each decision tree T actually builds a family of probability trees T_ω, indexed by $\omega \in \Omega$.

Lemma 1.1: *Let Ω be a non-empty set and let $T=(G,r)$, $G=(V,E)$, be a decision tree with decision map δ. Let $(w_u)_{u\in V}$ be the associated family of weight functions. Then the following holds:*

(a) *For all $u \in V$ and all $\omega \in \Omega$ we have $w_u(\omega) \geq 0$.*

(b) *For an arbitrary $u \in V$ let $\mathrm{ch}(u)$ denote the (finite) set of children of u. Then we have for all $\omega \in \Omega$*

$$\sum_{v\in \mathrm{ch}(u)} w_v(\omega) = w_u(\omega) \qquad \text{if } \mathrm{ch}(u) \neq \emptyset.$$

(c) *Let $\mathscr{L}_T$ denote the set of leaves of T. Then we have for all $\omega \in \Omega$:*

$$\sum_{v\in \mathscr{L}_T} w_v(\omega) = 1.$$

Proof. We fix an arbitrary $\omega \in \Omega$.

(a) If a vertex u has no parent, then $w_u(\omega) = 1 \geq 0$. If u has a parent v then we may assume by induction that $w_v(\omega) \geq 0$ and thus $w_u(\omega) = w_v(\omega) \cdot \delta(\omega,(v,u)) \geq 0$ because $\delta(\omega,(v,u)) \geq 0$ by definition.

(b) From the definition of w_v for each $v \in \mathrm{ch}(u)$ we get:

$$w_v(\omega) = w_u(\omega) \cdot \delta(\omega,(u,v)),$$

and thus it follows:

$$\sum_{v\in\mathrm{ch}(u)} w_v(\omega) = \sum_{v\in\mathrm{ch}(u)} w_u(\omega)\cdot\delta(\omega,(u,v)) = w_u(\omega)\cdot\sum_{v\in\mathrm{ch}(u)}\delta(\omega,(u,v)) = w_u(\omega)$$

because $\sum_{v\in\mathrm{ch}(u)} \delta(\omega,(u,v)) = 1$ according to the definition of δ.

(c) We proceed by induction over the height h of the tree T. For $h = 0$ we have exactly the one node r which is root and leaf, and we have

$$\sum_{v\in\mathscr{L}_T} w_v(\omega) = w_r(\omega) = 1$$

for all $\omega \in \Omega$ by definition. For the induction step let T_{h-1} denote the subtree generated by the vertices in the levels up to $h-1$, i.e. the tree where the leaves of the last level h and the corresponding edges are removed. The definition of the associated weights shows that for all $u \in T_{h-1}$ the weights coincide with those associated with T. The induction hypothesis yields then

$$\sum_{u\in\mathscr{L}_{T_{h-1}}} w_u(\omega) = 1.$$

We have a partition of $\mathscr{L}_{T_{h-1}}$ into two sets

$$\mathscr{L}_1 := \left\{u \in \mathscr{L}_{T_{h-1}} \;\middle|\; \mathrm{ch}(u) = \emptyset \text{ in } T\right\} \qquad \text{and} \qquad \mathscr{L}_2 := \left\{u \in \mathscr{L}_{T_{h-1}} \;\middle|\; \mathrm{ch}(u) \neq \emptyset \text{ in } T\right\}.$$

The leaves in $\mathscr{L}_T$ are then given by the disjoint union

$$\mathscr{L}_T = \mathscr{L}_1 \,\dot\cup \dot{\bigcup}_{u\in\mathscr{L}_2} \mathrm{ch}(u).$$

From this it follows that

$$\begin{aligned}\sum_{u\in\mathscr{L}_T} w_u(\omega) &= \sum_{u\in\mathscr{L}_1} w_u(\omega) + \sum_{u\in\mathscr{L}_2}\sum_{v\in\mathrm{ch}(u)} w_v(\omega)\\ &= \sum_{u\in\mathscr{L}_1} w_u(\omega) + \sum_{u\in\mathscr{L}_2} w_u(\omega)\\ &= \sum_{u\in\mathscr{L}_{T_{h-1}}} w_u(\omega) = 1\end{aligned}$$

where we have used **(b)** with

$$\sum_{v\in\mathrm{ch}(u)} w_v(\omega) = w_u(\omega)$$

for each $u \in \mathscr{L}_2$ and each $\omega \in \Omega$. □

As an immediate consequence of this lemma we get:

Theorem 1.2: *Let Ω be a non-empty set and let $T = (G,r)$, $G = (V,E)$, be a decision tree with decision map δ. Let $(w_u)_{u\in V}$ be the associated family of weight functions. Then the family $(w_u)_{u\in\mathscr{L}_T}$, i.e. the family of weight functions associated with the leaves of T, provides a normalized family of weight functions.* □

We call this family $(w_u)_{u\in\mathscr{L}_T}$ the ***decision tree based weight functions*** on Ω induced by the decision tree T and the decision map δ.

To be able to write the weight functions in a different way for binary decision trees, we need the complement operator

$$c : [0,1] \longrightarrow [0,1], \quad x \longmapsto 1-x.$$

It is strongly monotonously decreasing, i.e.

$$x < y \quad \Longrightarrow \quad c(x) > c(y) \qquad \text{for all } x,y \in [0,1],$$

and involutory, i.e.

$$c(c(x)) = x \qquad \text{for all } x \in [0,1].$$

By the latter property it follows (using operator notation):

$$c^n(x) = \begin{cases} \mathrm{Id}(x) = x & n \in \mathbb{N},\ n \text{ even}, \\ c(x) = 1-x & n \in \mathbb{N},\ n \text{ odd}, \end{cases}$$

where we set, as usual, $c^0(x) := \mathrm{Id}(x) := x$, and recursivley $c^n(x) = c(c^{n-1}(c))$ for $n \in \mathbb{N} \setminus \{0\}$.

Let $u \in \mathscr{L}_T$ be a leaf of a binary decision tree $T = (G,r)$, $G = (V,E)$, and $(u_0,\ldots,u_n)$ with $u_0 = r$ and $u_n = u$ the path from the root r to the leaf u. Then it is easy to see by induction that

$$w_u(\omega) = \sum_{i=0}^{n-1} c^{q_{u_i}(u_{i+1})}\delta_{u_i}(\omega)$$

where $q_u : V \longrightarrow \{0,1\}$ denotes for each node $u \in V$ the enumeration of the (binary) decision tree T.

Examples of decision tree based weight functions We proceed now with special realizations of families of decision tree based weight functions in the case of a binary decision tree T. We only need to define the components of the decision map, i.e. for each inner node $u \in \mathscr{N}_T$ we have to fix

$$\delta_u : \Omega \longrightarrow [0,1].$$

Examples: 1. Choose $\Omega := \mathbb{R}^d$ for some $d \in \mathbb{N}$, and assign to each inner node $u \in \mathscr{L}_T$ a vector $\alpha_u \in \mathbb{R}^d$ and a real number $\beta_u \in \mathbb{R}$. Then a ***splitting rule*** is defined by

$$\delta_u(\omega) := (\alpha_u^\top \omega > \beta_u) \qquad \text{for all } \omega \in \Omega$$

where we interpret $>$ as a binary operator $> : \mathbb{R}\times\mathbb{R} \longrightarrow \{0,1\}$ with values 0 (`FALSE`) and 1 (`TRUE`). The resulting decision tree based weight functions are then the characteristic functions describing a partition of $\Omega = \mathbb{R}^d$ into a finite set of polyhedra.

2. If in the previous example we choose especially the vectors α_u to be unit vectors,

$$\alpha_u \in \left\{ e_i \;\middle|\; i = 1, \ldots, d \right\}$$

with

$$e_i \in \mathbb{R}^d, \qquad (e_i)_j := \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{else,} \end{cases}$$

then we get a partition of Ω into hypercuboids with axis-parallel edges.

3. We may also „fuzzyficate“ the operator $>$ to get smoother weight functions. Following Barrena Algara [2007], we do this by first choosing a sigmoid function $s : \mathbb{R} \longrightarrow [0,1]$, i.e. s is a monotonously increasing and continuous function with

$$s(x) \to 0 \text{ for } x \to -\infty \qquad \text{and} \qquad s(x) \to 1 \text{ for } x \to +\infty.$$

We then define

$$\delta_u(\omega) := s(\alpha_u^\top \omega - \beta_u).$$

We may especially choose the ***logistic function***

$$s : \mathbb{R} \longrightarrow [0,1], \qquad s(x) := \frac{1}{1+e^{-x}}.$$

This function is well-known as the solution of the so-called logistic differential equation or as the usual activity function for neural networks. For the logistic function, the additional property

$$1 - s(x) = s(-x)$$

holds, and the decision map is thus given by

$$\delta(\omega, (u, v_1)) = \delta_u(\omega) = s(\alpha_u^\top \omega - \beta_u)$$

and

$$\delta(\omega, (u, v_0)) = 1 - \delta_u(\omega) = s(\beta_u - \alpha_u^\top \omega).$$

The decision tree based weight functions presented in this section are based on an idea of Eva Barrena [Barrena Algara, 2007]. Her idea is the application of so-called fuzzy-less and fuzzy-greater operators in the splitting rules occurring in decision trees. Barrena uses these modified decision trees, called ***soft operator decision trees (SODT)***, for the purpose of classification: exactly the task decision trees have been invented for. In her thesis, Eva Barrena investigates the improvements of SODTs over the usual “crisp” decision trees with hard splitting rules. She already defines what we call probability trees (there: possibility trees), and also the weight functions we called decision tree based weight functions. Here, in this thesis, we want to use her weight functions for local model networks and thus in the LOLIMOT algorithm. All proofs concerning probability trees and decision tree based weight functions can already be found in Eva Barrena’s thesis. We have imbedded them in our framework, trying to unify these concepts with the original concepts used in local model networks and the LOLIMOT algorithm on one side, and on the other side to reveal thus the differences.

Smooth decision tree based weight functions for the LOLIMOT algorithm

As already mentioned, the normalized Gaussian weight functions originally used for local model networks and the LOLIMOT algorithm have some sever disadvantages.

Reactivation One of the disadvantages is the uncertainty about the behaviour outside the prescribed data area. Here, the normalization of the Gaussian weight functions causes effects which are called reactivation. That means, one of the weight functions reappears unexpectedly in some region far outside the data area. To avoid this, Nelles [1997] proposes to freeze the values given at the borders of the data area when going outside. The phenomenon already appears in the 1-dimensional case with two unnormalized weight functions

$$w_i = \exp\Big(-\frac{1}{2}\frac{(x-\mu_i)^2}{\sigma_i^2}\Big), \qquad i = 1,2,$$

with $\mu_1 \neq \mu_2$, $\sigma_1 \neq \sigma_2$ and $\sigma_1, \sigma_2 > 0$. If we assume e.g. $\mu_1 < \mu_2$, then w_1 constitutes the left and w_2 the right regime. One would expect that for all $x < \mu_1$ the weight w_1 dominates w_2, i.e. $w_1(x) > w_2(x)$, and that on the other side for $x > \mu_2$, we would have $w_1(x) < w_2(x)$. We will show that this is not possible: If we take a look at those points $x \in \mathbb{R}$ where w_1 and w_2 are equally valid, i.e.

$$\exp\Big(-\frac{1}{2}\frac{(x-\mu_1)^2}{\sigma_1^2}\Big) = \exp\Big(-\frac{1}{2}\frac{(x-\mu_2)^2}{\sigma_2^2}\Big),$$

then from the bijectivity of the exponential function it follows that this is equivalent to

$$\frac{(x-\mu_1)^2}{\sigma_1^2} = \frac{(x-\mu_2)^2}{\sigma_2^2},$$

and this in turn is equivalent to

$$q(x) := \big(\sigma_1(x-\mu_2)\big)^2 - \big(\sigma_2(x-\mu_1)\big)^2 = 0.$$

We thus have to determine the zeros of the quadratic polynomial $q(x)$. They are given by

$$x_1 = \frac{\sigma_1\mu_2 - \sigma_2\mu_1}{\sigma_1 - \sigma_2} \qquad \text{and} \qquad x_2 = \frac{\sigma_1\mu_2 + \sigma_2\mu_1}{\sigma_1 + \sigma_2}.$$

They exist because $\sigma_1 \neq \sigma_2$ and $\sigma_1, \sigma_2 > 0$. They are not equal, because from

$$x_2 - x_1 = \frac{(\sigma_1-\sigma_2)(\sigma_1\mu_2+\sigma_2\mu_1) - (\sigma_1+\sigma_2)(\sigma_1\mu_2-\sigma_2\mu_1)}{\sigma_1^2-\sigma_2^2} = \frac{2\sigma_1\sigma_2(\mu_1-\mu_2)}{\sigma_1^2-\sigma_2^2}$$

it is easily seen that $x_1 = x_2$ if and only if $\mu_1 = \mu_2$. Also, we see, under the assumption $\mu_1 < \mu_2$, that $x_1 < x_2$ if and only if $\sigma_1 < \sigma_2$.

The derivative $q'(x) := dq(x)/dx$ of the above polynomial is given by

$$q'(x) = 2\big((\sigma_1^2-\sigma_2^2)x - \sigma_1^2\mu_2 + \sigma_2^2\mu_1\big)$$

and inserting x_1 and x_2 results in

$$q'(x_1) = 2\sigma_1\sigma_2(\mu_2 - \mu_1) \neq 0$$

and

$$q'(x_2) = 2\sigma_1\sigma_2(\mu_1 - \mu_2) \neq 0,$$

respectively. Concluding, we see that at the points x_1 and x_2, and only at these points, the dominant weight changes.

The derivative of the weight functions w_i at the crossing points x_1 and x_2 are given by

$$\frac{d}{dx} w_i(x)\big|_{x=x_j} = w_i(x_j) \cdot \Big(- \frac{x_j - \mu_i}{\sigma_i^2} \Big),$$

and since $w_1(x_j) = w_2(x_j) > 0$, $j = 1,2$, we have that

$$\frac{d}{dx}(w_2(x) - w_1(x))\big|_{x=x_j} > 0$$

if and only if

$$-\frac{x_j - \mu_2}{\sigma_2^2} + \frac{x_j - \mu_1}{\sigma_1^2} > 0.$$

Since

$$-\frac{x_j - \mu_2}{\sigma_2^2} + \frac{x_j - \mu_1}{\sigma_1^2} = \begin{cases} \dfrac{\mu_1 - \mu_2}{\sigma_1\sigma_2} & \text{if } j = 1, \\ \dfrac{\mu_2 - \mu_1}{\sigma_1\sigma_2} & \text{if } j = 2, \end{cases}$$

we have that the sign of

$$\frac{d}{dx}(w_2(x) - w_1(x))\big|_{x=x_j}$$

depends only on the sign of $\mu_2 - \mu_1$: at x_1, we have the opposite sign, and at x_2, we have the same sign. In our case, the sign of $\mu_2 - \mu_1$ is positive, and we observe that at x_1 the dominance of the weight functions changes from $w_2(x) > w_1(x)$ for all $x < x_1$ to $w_2(x) < w_1(x)$ for all x with $x_1 < x < x_2$, and it changes again at x_2 to $w_2(x) > w_1(x)$ for all $x > x_2$. This is the reactivation of w_2 at the left side. (We considered the non-normalized weight functions, but the dominance does not change after normalization).

Other problems Another problem is the fast (exponential) convergence to zero outside of the data area of all weight functions, which finally brings all the weight functions numerically to zero. During normalization, a division by zero occurs. But even inside the data area, the normalized Gaussian weight functions may show unexpected shapes: there occur bumps and curls in the overlapping areas. Another disadvantage is the impossibility to rotate the Gaussian functions around their middlepoints without overlapping the neighbouring areas. This overlapping would even aggravate the former mentioned problems. So it is practically not easy to obtain areas with oblique borders.

To avoid all these problems, we apply the decision tree based weight functions as another kind of weight functions for the LOLIMOT algorithm. It is not necessary to normalize them, because they already build a partition of one through their construction. So, all the problems caused by normalization do not appear.

To use this kind of weight functions for the LOLIMOT algorithm, we just have to compute them via the binary decision trees constructed during the run of the LOLIMOT algorithm. It is clear that we thus get only rules of the form $\alpha_u^\top \omega > \beta_u$ with $\alpha_u = e_i$ for some unit vector e_i, or at most $\alpha_u = \sigma e_i$ for some chosen positive constant σ as in the original algorithm. We will provide some possibilities to loosen these restrictions at the end of the present chapter.

In figure 1.11 a comparison of the different weight functions is shown through colour plots with an example.

Interpretation as Fuzzy rules If, as in the second example of the previous section, $\alpha_u = e_i$ for some unit vector e_i, then, the decision tree based weight functions can be directly translated into Fuzzy rules using so-called Fuzzy operators. Following Barrena Algara [2007], we define the Fuzzy operator $>_s$ by

$$>_s \colon \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}, \quad x >_s y := s(x-y)$$

with

$$s(z) = \frac{1}{1+e^{-z}}.$$

Interpreting `AND` as multiplication and `NOT` x as $1-x$ and setting $x > y := x >_s y$ and $x < y := \texttt{NOT}(x > y) := 1 - (x >_s y)$ for all $x, y \in \mathbb{R}$, we can for example transform the weighted basis function

$$w(\psi_1, \psi_2) \cdot f(\psi_1, \psi_2)$$

with

$$\begin{aligned} w(\psi_1, \psi_2) &= (1 - (\psi_1 >_s 0.5)) \cdot (\psi_2 >_s 0.5) \cdot (\psi_1 >_s 0.25) \\ &= (1 - s(\psi_1 - 0.5)) s(\psi_2 - 0.5) s(\psi_1 - 0.25) \end{aligned}$$

into the Fuzzy-rule

$$\texttt{IF NOT } \psi_1 > 0.5 \texttt{ AND } \psi_2 > 0.5 \texttt{ AND } \psi_1 > 0.25 \texttt{ THEN } f(\psi_1, \psi_2)$$

(see figure 1.12; the weight function corresponds to the area `C` and yields the third Fuzzy rule). One could as well reverse the procedure.

Generalization to oblique borders Giving up the restriction $\alpha_u = e_i$, we get areas with oblique borders. By replacing for example the first condition in the weight function considered above, i.e.

$$\begin{aligned} w(\psi_1, \psi_2) &= (1 - (\psi_1 >_s 0.5)) \cdot (\psi_2 >_s 0.5) \cdot (\psi_1 >_s 0.25) \\ &= (1 - s(\psi_1 - 0.5)) s(\psi_2 - 0.5) s(\psi_1 - 0.25), \end{aligned}$$

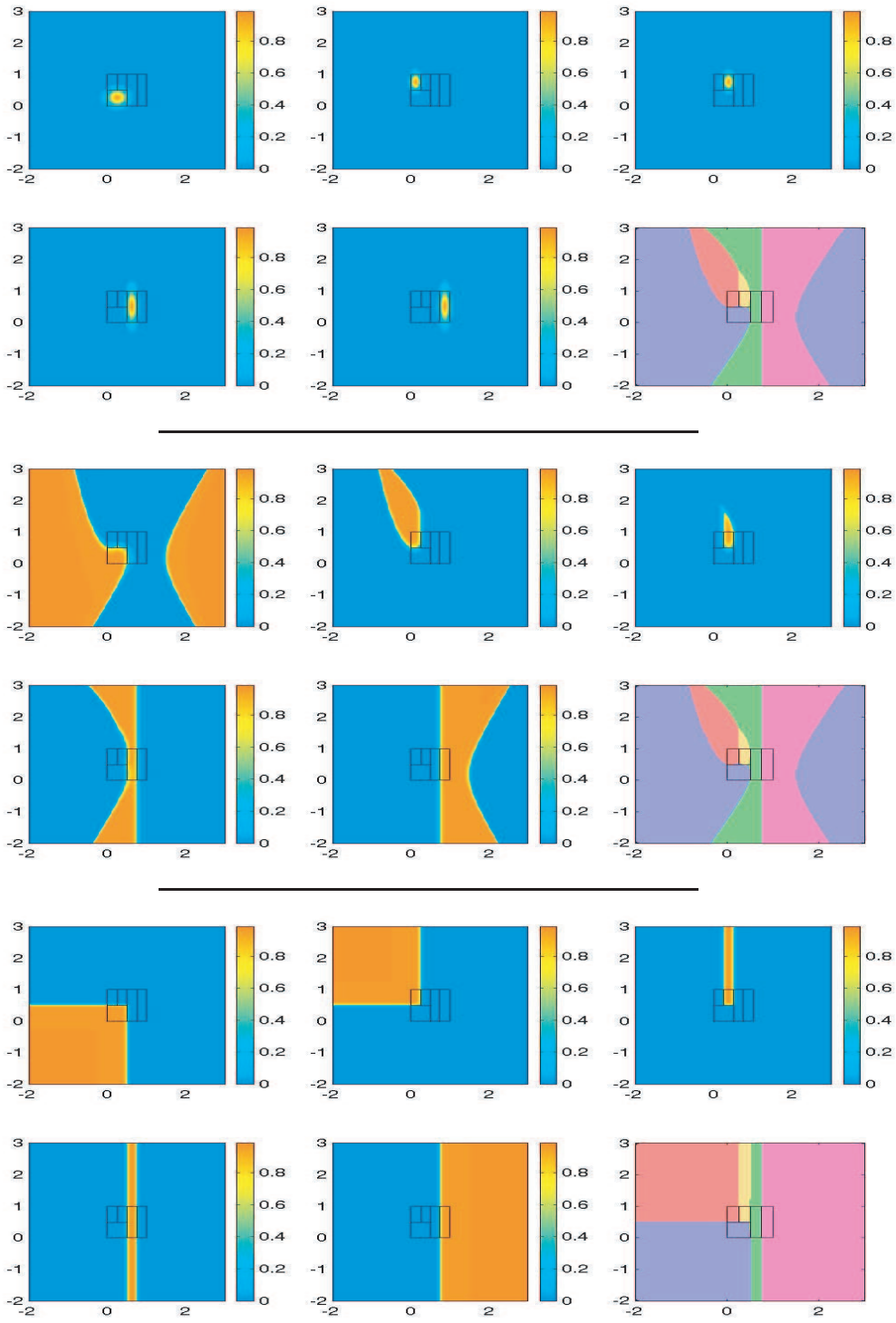


Figure 1.11: Weight functions: Gauss (top), Normalized Gauss (middle), Decision Tree Based (bottom); shown are respectively the weight functions of each of the five regimes, and (in the respective lower right corner) the dominant regimes

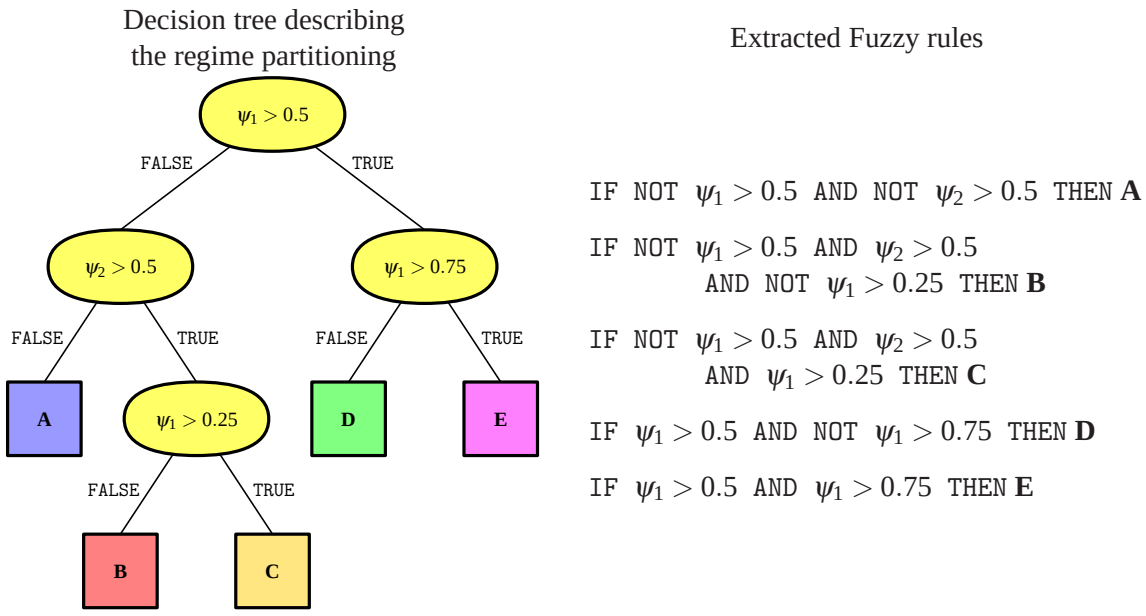


Figure 1.12: Extraction of Fuzzy rules from a decision tree

we still could write something like

$$\texttt{IF NOT } \alpha_1\psi_1 + \alpha_2\psi_2 > \beta \texttt{ AND } \psi_2 > 0.5 \texttt{ AND } \psi_1 > 0.25 \texttt{ THEN } f(\psi_1, \psi_2),$$

but this contradicts the philosophy of Fuzzy theory to treat each dimension separately.

1.4.2 Gradient based optimization

We develop in this section the application of a gradient based local optimization method for the local model networks, in addition to the constructive method realized by the LOLIMOT algorithm. The local optimization methods always require starting values. The parameters constructed by the LOLIMOT algorithm may serve to this purpose. The idea of an iterative after-optimization of the local model network obtained by the LOLIMOT algorithm appears already in Nelles [1997].

Problem formulation The problem we want to solve is an optimization problem: Let Ω be a set and $f : \Omega \longrightarrow \mathbb{R}$ be a function, then we want to find a value $\omega^* \in \Omega$ such that $f(\omega^*)$ is minimal (or maximal) with respect to all values $f(\omega)$, $\omega \in \Omega$. We will restrict our investigations to the case that we want to find a minimum. The maximum can be found by the same method using the function $-f$ instead of f. If f is a complicated function, then this problem is very difficult. An easier problem is to find a local minimum when $\Omega := \mathbb{R}^n$ and the function f is differentiable with respect to $\omega \in \Omega$. Gradient based optimization starts with an initial value ω_0 and uses the gradient of f to find the direction pointing downwards. The initial value ω_0 is updated to the value ω_1 by making a step towards this direction. These steps are repeated, always going downward, thus producing a sequence $\omega_0, \omega_1, \omega_2, \ldots$ which (under

certain conditions concerning the size of the steps) converges to the next local minimum. The method thus pictured uses the steepest descent direction pointing against the gradient. There are better choices of directions when information about the curvature of the function f around the points ω_i is taken into account. This leads to the Newton and Quasi-Newton directions (explained in detail later).

Different initial values ω_0 will lead to different local minima, and it should be clear that we never can be sure that we have found the global minimum. Nevertheless, this method is *the* method used for the „training“ of neural networks, known as backpropagation. As mentioned, we always need an initial value; with neural networks, the initial value is mostly chosen randomly. Surprisingly, in the neural network literature the important step of looking for the right distribution of these random values is rarely addressed, although one has necessarily to decide for one distribution when implementing the algorithm. Another surprising fact is that the backpropagation algorithm is in praxis often implemented using the steepest descent method, resulting in an unnecessarily slow convergence. It can be fastened without much effort using Quasi-Newton steps. Here, also ideal step sizes are known. We will describe this in more detail in the following sections. Concretely spoken, we choose the Levenberg-Marquardt descent direction and the Armijo step size.

In our case, the arguments subject to optimization are the parameters of our model. Since the local model networks obtained by the LOLIMOT algorithm are designed in a way such that they are differentiable with respect to the parameters, it is possible to apply a gradient based algorithm. Our function f is in this case an error function measuring the error between model and real system. The derivatives of this error function will be computed by means of the chain rule. In contrast to the usual backpropagation algorithm of neural networks we apply a feedforward computation. This is necessary because our overall approach is an output error (NOE) approach better suited to simulation purposes, see Nerrand et al. [1993]. In the mentioned article, the algorithm we use is called undirected algorithm.

Identification set and validation set The error function compares computed output to measured output that is necessarily noisy and disturbed. To avoid that the parameters are too much adapted to this noise (in neural network theory this is called overfitting), we have to take some precautions. This is done by dividing the set of measurements $\mathscr{E}$ into three disjoint sets, the ***identification set*** (training set) $\mathscr{E}^{\text{Ident}}$, the ***validation set*** $\mathscr{E}^{\text{Valid}}$, and the ***test set*** $\mathscr{E}^{\text{Test}}$, respectively. The identification set serves to the estimation of the parameters, whereas the validation set is used to check against a too strong fitting of the optimized parameters to the data of the identification set. The test set will be untouched until after the identification to test the quality of the identified model.

If we denote the cardinality of the sets $\mathscr{E}^{\text{Ident}}$ and $\mathscr{E}^{\text{Valid}}$ by $N := \#\mathscr{E}^{\text{Ident}}$ and $M := \#\mathscr{E}^{\text{Valid}}$, respectively, we are able to compute the difference between measured (observed) and computed (estimated) data with respect to each of the sets $\mathscr{E}^{\text{Ident}}$ and $\mathscr{E}^{\text{Valid}}$ by means of quadratic error functions. Thus, for all parameters $\theta \in \Theta$ we define with respect to the identification set $\mathscr{E}^{\text{Ident}}$ the ***identification error***

$$V_N(\theta) := \frac{1}{2N} \sum_{(u,y)\in\mathscr{E}^{\text{Ident}}} \left| y - \Gamma(u;\theta) \right|^2$$

and with respect to the validation set $\mathscr{E}^{\text{Valid}}$ the ***validation error***

$$W_M(\theta) := \frac{1}{2M} \sum_{(u,y)\in\mathscr{E}^{\text{Valid}}} \left|y - \Gamma(u;\theta)\right|^2.$$

In both cases Γ denotes the parameterized model. The optimization algorithm changes the parameters θ in such a way that the identification error $V_N(\theta)$ decreases with every optimization step. The observation of the validation error $W_M(\theta)$ guarantees that no overfitting of the parameters to the identification data taken from $\mathscr{E}^{\text{Ident}}$ occurs. Because of the bias-variance decomposition of the complete expected error (see e.g. Sjöberg et al. [1994], section 6.3) an overfitting of the parameters θ can therefore be detected by a beginning increase of the validation error $W_M(\theta)$. The observation of the error plot helps to decide when the optimization procedure should be stopped.

Application to local model networks

We give now a detailed description of the application of the gradient based optimization for local model networks.

Given data Let $\mathscr{E}^{\text{Ident}} = \{(u_1,y_1),\ldots,(u_N,y_N)\}$ be a finite set of input/output measurements of the real system taken at times $t_1 < \ldots < t_N$, respectively. These measurements may as well come from several, say r, experiments from the real system, starting at times

$$t_0^{(1)},\ldots,t_0^{(r)} \in \mathscr{T},$$

i.e.

$$(t_1,\ldots,t_N) = (t_0^{(1)},\ldots,t_{l_1-1}^{(1)}, \quad t_0^{(2)},\ldots,t_{l_2-1}^{(2)}, \quad \ldots, \quad t_0^{(r)},\ldots,t_{l_r-1}^{(r)})$$

where $l_1,\ldots,l_r$ are the length of the experiments. For a given parameter vector

$$\theta := (\theta_1,\ldots,\theta_d)^\top \in \Theta = \mathbb{R}^d$$

let further be

$$\hat{y}_k(\theta) := \Gamma(u(\cdot);\theta)(t_k) := \Gamma_{t_0^{(\rho(k))},x_0^{(\rho(k))}}(u(\cdot);\theta)(t_k), \quad k = 1,\ldots,N,$$

the corresponding output of the parameterized model where the initial times $t_0^{(\rho(k))}$ and initial values $x_0^{(\rho(k))}$ are with respect to the index $\rho(k) \in \{1,\ldots,r\}$ being the maximal ρ such that

$$t_0^{(\rho)} \le t_k.$$

Gradient of the error function We define the following error function depending on the parameters $\theta \in \Theta$:

$$V_N(\theta) := \frac{1}{2N} \sum_{k=1}^{N} \left| y_k - \hat{y}_k(\theta) \right|^2 = \frac{1}{2N} \sum_{k=1}^{N} \left| y_k - \Gamma(u(\cdot);\theta)(t_k) \right|^2.$$

The derivative of this error function $V_N(\theta)$ with respect to the parameter vector θ is computed by

$$V_N'(\theta) := \frac{d}{d\theta} V_N(\theta) = -\frac{1}{N} \sum_{k=1}^{N} \big(y_k - \Gamma(u(\cdot);\theta) \big)(t_k) \frac{\partial}{\partial \theta} \Gamma(u(\cdot);\theta)(t_k).$$

Essential for the computation of this derivative therefore is the computation of the gradient of the model with respect to θ, i.e. we need to compute for all $k = 1, \ldots, N$ the $(1 \times d)$ vectors

$$\frac{\partial}{\partial \theta} \Gamma(u(\cdot);\theta)(t_k) = \left(\frac{\partial}{\partial \theta_1} \Gamma(u(\cdot);\theta)(t_k), \ldots, \frac{\partial}{\partial \theta_d} \Gamma(u(\cdot);\theta)(t_k) \right).$$

As an abbreviation for these gradients we denote them by

$$\psi_k(\theta) := \frac{\partial}{\partial \theta} \Gamma(u(\cdot);\theta)(t_k) \quad \text{for all } k = 1, \ldots, N$$

and their components by

$$\psi_{k,j}(\theta) := \frac{\partial}{\partial \theta_j} \Gamma(u(\cdot);\theta)(t_k) \quad \text{for all } k = 1, \ldots, N \text{ and } j = 1, \ldots, d.$$

We fix now a $k \in \{1, \ldots, N\}$ and consider the computation of the components $\psi_{k,j}(\theta)$ for all $j = 1, \ldots, d$. According to the definition of a local model network, our model $\Gamma(u(\cdot);\theta)(t_k)$ is given by

$$\hat{y}(\theta) = \Gamma(u(\cdot);\theta)(t_k) = \sum_{i=1}^{N} w_i(x(t_k);\theta) \eta_i(x(t_k);\theta) = \sum_{i=1}^{N} w\big(x(t_k);\theta_i^w\big)\, \eta\left(x(t_k);\theta_i^\eta\right)$$

resulting in the derivative

$$\begin{aligned} \psi_{k,j}(\theta) = \frac{\partial}{\partial \theta_j} \Gamma(u(\cdot);\theta)(t_k) &= \frac{\partial}{\partial \theta_j} \sum_{i=1}^{N} w\big(x(t_k);\theta_i^w\big)\, \eta\left(x(t_k);\theta_i^\eta\right) \\ &= \sum_{i=1}^{N} \left[\frac{\partial w\big(x(t_k);\theta_i^w\big)}{\partial \theta_j} \cdot \eta\left(x(t_k);\theta_i^\eta\right) + w\big(x(t_k);\theta_i^w\big) \cdot \frac{\partial \eta\left(x(t_k);\theta_i^\eta\right)}{\partial \theta_j} \right] \end{aligned}$$

for all $j = 1, \ldots, d$.

Gradient of partial models We continue with the derivative of the partial models

$$\eta\left(x(t_k);\theta_i^{\eta}\right) = {\theta_i^{\eta}}^{\top} x(t_k)$$

with linear parameters θ_i^{η}. It is given by

$$\frac{\partial \eta\left(x(t_k);\theta_i^{\eta}\right)}{\partial \theta_j} = \frac{\partial {\theta_i^{\eta}}^{\top} x(t_k)}{\partial \theta_j} = \left(\frac{\partial \theta_i^{\eta}}{\partial \theta_j}\right)^{\top} x(t_k) + {\theta_i^{\eta}}^{\top} \frac{\partial x(t_k)}{\partial \theta_j}$$

with

$$\left(\frac{\partial \theta_i^{\eta}}{\partial \theta_j}\right)^{\top} x(t_k) = \begin{cases} x(t_k), & \text{if } (\theta_i^{\eta})_l \equiv \theta_j, \\ 0, & \text{else,} \end{cases}$$

where $(\theta_i^{\eta})_l$ for some $l \in \mathbb{N}$ shall denote a component of the vector θ_i^{η}. Concerning

$$\partial x(t_k)/\partial \theta_j,$$

we have to be aware that also the state vector $x(t_k)$ at time t_k may depend on θ_j. If we choose ARX models as partial models, i.e.

$$x(t) = \big(u(t), u(t-1), \ldots, u(t-n_u), y(t-1), \ldots, y(t-n_y)\big)^{\top},$$

this is not the case, and the derivative $\partial x(t_k)/\partial \theta_j$ is always equal to 0. In contrast, with OE models as partial models, i.e.

$$x(t) = \big(u(t), u(t-1), \ldots, u(t-n_u), \hat{y}(t-1;\theta), \ldots, \hat{y}(t-n_y;\theta)\big)^{\top},$$

the corresponding derivative is given by

$$\frac{\partial x(t_k)}{\partial \theta_j} = \left(0, 0, \ldots, 0, \frac{\partial \hat{y}(t_k-1;\theta)}{\partial \theta_j}, \ldots, \frac{\partial \hat{y}(t_k-n_y;\theta)}{\partial \theta_j}\right)^{\top}.$$

The derivatives

$$\frac{\partial \hat{y}(\tau;\theta)}{\partial \theta_j} \quad \text{for } \tau < t_k$$

can be recursively computed, as long as $\tau \geq t_0^{(\rho)}$ if $t_k = t_\kappa^{(\rho)}$ for some $1 \leq \rho \leq r$ and some $0 \leq \kappa < l_\rho$; one has to provide the initial values

$$\frac{\partial \hat{y}(\tau;\theta)}{\partial \theta_j} \quad \text{for } \tau \leq t_0^{(\rho)}$$

for the other cases.

Gradient of weight functions The derivation of the weight functions is a bit more involved. We have two possibilities: the original weight functions consisting of normalized Gaussian bell clocks, and the decision tree based weight functions. We look at the two possibilities separately.

- **Possibility 1**: We may think of the original weight functions as being assigned to the leaves of a given decision tree $T = (G, r)$. If for every leaf $u \in \mathscr{L}_T$ the non-normalized weight is denoted by $\tilde{w}_u(t;\theta^w)$, then the normalization procedure yields

$$w_u(t;\theta^w) = \frac{\tilde{w}_u(t;\theta^w)}{\sum_{v\in\mathscr{L}_T}\tilde{w}_v(t;\theta^w)}.$$

By assuming differentiability of all non-normalized weight functions $\tilde{w}_u(t;\theta^w)$ with respect to θ^w, we obtain the derivative of the normalized weight functions by:

$$\frac{\partial}{\partial\theta^w}w_u(t;\theta^w) = \frac{\left(\frac{\partial}{\partial\theta^w}\tilde{w}_u(t;\theta^w)\right)\cdot\sum_{v\in\mathscr{L}_T}\tilde{w}_v(t;\theta^w) - \tilde{w}_u(t;\theta^w)\cdot\sum_{v\in\mathscr{L}_T}\frac{\partial}{\partial\theta^w}\tilde{w}_v(t;\theta^w)}{\left(\sum_{v\in\mathscr{L}_T}\tilde{w}_v(t;\theta^w)\right)^2}.$$

In particular, we choose as non-normalized weight functions the Gaussian bell functions

$$\tilde{w}_u(t;\theta^w) := \exp\left(-\frac{1}{2}(x(t)-\mu_u)^\top\Sigma_u(x(t)-\mu_u)\right)$$

(we suppress the usual constant factor which is without any relevance due to the normalization procedure) with $\mu_u \in \mathbb{R}^d$ and a symmetric positive definite matrix $\Sigma_u \in \mathbb{R}^{d\times d}$, together building the parameters θ^w. Since Σ_u is positive definite, the square root $\Sigma^{1/2} \in \mathbb{R}^{d\times d}$ exists and is also symmetric and positive definite. We therefore write

$$x_u(t;\theta^w) := \Sigma_u^{1/2}(x(t)-\mu_u)$$

which leads to

$$\tilde{w}_u(t;\theta^w) = \exp\left(-\frac{1}{2}x_u(t;\theta^w)^\top x_u(t;\theta^w)\right).$$

As derivative with respect to θ^w we obtain

$$\begin{aligned}\frac{\partial}{\partial\theta^w}\tilde{w}_u(t;\theta^w) &= -\exp\left(-\frac{1}{2}x_u(t;\theta^w)^\top x_u(t;\theta^w)\right)x_u(t;\theta^w)^\top\frac{\partial}{\partial\theta^w}x_u(t;\theta^w)\\ &= -\tilde{w}_u(t;\theta^w)x_u(t;\theta^w)^\top\frac{\partial}{\partial\theta^w}x_u(t;\theta^w).\end{aligned}$$

Inserting this into the above given derivative of the normalized weight function $w_u(t;\theta^w)$,

we obtain

$$\begin{aligned}\frac{\partial}{\partial\theta^w}w_u(t;\theta^w) &= \frac{\left(-\tilde{w}_u(t;\theta^w)x_u(t;\theta^w)^\top\frac{\partial}{\partial\theta^w}x_u(t;\theta^w)\right)\cdot\sum_{v\in\mathscr{L}_T}\tilde{w}_v(t;\theta^w)}{\left(\sum_{v\in\mathscr{L}_T}\tilde{w}_v(t;\theta^w)\right)^2}\\ &\quad-\frac{\tilde{w}_u(t;\theta^w)\cdot\sum_{v\in\mathscr{L}_T}(-\tilde{w}_v(t;\theta^w))x_v(t;\theta^w)^\top\frac{\partial}{\partial\theta^w}x_v(t;\theta^w)}{\left(\sum_{v\in\mathscr{L}_T}\tilde{w}_v(t;\theta^w)\right)^2}\\ &= w_u(t;\theta^w)\cdot\Bigg[-x_u(t;\theta^w)^\top\frac{\partial}{\partial\theta^w}x_u(t;\theta^w)\\ &\qquad+\sum_{v\in\mathscr{L}_T}w_v(t;\theta^w)x_v(t;\theta^w)^\top\frac{\partial}{\partial\theta^w}x_v(t;\theta^w)\Bigg].\end{aligned}$$

We find also:

$$\frac{\partial x_u(t;\theta^w)}{\partial\mu_u}=\Sigma_u^{1/2}\left(\frac{\partial x(t)}{\partial\mu_u}-1\right)\qquad\text{and}\qquad\frac{\partial x_u(t;\theta^w)}{\partial\Sigma_u^{1/2}}=\Sigma_u^{1/2}\frac{\partial x(t)}{\partial\Sigma_u^{1/2}}+x(t)-\mu_u,$$

where $x(t)$ may depend on the parameters, as mentioned earlier.

- **Possibility 2**: We use the weight functions associated with a given full binary decision tree $T=(G,r)$, $G=(V,E)$, with the parameterized components of the normalized edge weights given by $\delta_u(\cdot;\theta^w)$ for each inner vertex $u\in\mathscr{N}_T$. We have seen that the weight function for a leaf $u\in\mathscr{L}_T$ can be written as

$$w_u(t;\theta^w)=\sum_{i=0}^{n-1}c^{q_{u_i}(u_{i+1})}\delta_{u_i}(t;\theta^w)$$

where $(u_0,\ldots,u_n)$ with $u_0=r$ and $u_n=u$ denotes the path from the root r to the leaf u, where q_u denotes for each node $u\in V$ the enumeration of the (binary) decision tree T, where

$$c:[0,1]\longrightarrow[0,1],\quad x\longmapsto 1-x,$$

and where δ_u are the components of a decision map for T. By assuming again differentiability of the components $\delta_u(t;\theta^w)$ with respect to θ^w, we obtain for the derivative of the weight function w_u for a leaf $u\in\mathscr{L}_T$, with $(u_0,\ldots,u_n)$ being the path from r to u, as:

$$\begin{aligned}\frac{\partial}{\partial\theta^w}w_u(t;\theta^w)&=\sum_{i=0}^{n-1}\frac{\partial}{\partial\theta^w}c^{q_{u_i}(u_{i+1})}(\delta_{u_i}(t;\theta^w))\cdot\prod_{\substack{j=0\\j\neq i}}^{n-1}c^{q_{u_j}(u_{j+1})}(\delta_{u_j}(t;\theta^w))\\&=\sum_{i=0}^{n-1}(-1)^{q_{u_i}(u_{i+1})}\frac{\partial}{\partial\theta^w}\delta_{u_i}(t;\theta^w)\cdot\prod_{\substack{j=0\\j\neq i}}^{n-1}c^{q_{u_j}(u_{j+1})}(\delta_{u_j}(t;\theta^w)).\end{aligned}$$

In particular, in this case we may choose the logistic function

$$\delta_u(t;\theta^w) := s(\alpha_u^\top x(t) - \beta_u) = \frac{1}{1+\exp(-\alpha_u^\top x(t) + \beta_u)}$$

where $\alpha_u \in \mathbb{R}^d$ and $\beta_u \in \mathbb{R}$ constitute the parameters θ^w. If we define

$$x_u(t;\theta^w) := \alpha_u^\top x(t) - \beta_u,$$

we can write this as

$$\delta_u(t;\theta^w) = s(x_u(t;\theta^w)) = \frac{1}{1+\exp(-x_u(t;\theta^w))}.$$

Since

$$s'(x) := \frac{\partial}{\partial x} s(x) = s(x)(1-s(x)) = s(x)s(-x),$$

the derivative of δ_u with respect to θ^w is in this case given by

$$\begin{aligned}
\frac{\partial}{\partial \theta^w}\delta_u(t;\theta^w) &= s(x_u(t;\theta^w))(1-s(x_u(t;\theta^w)))\frac{\partial}{\partial \theta^w}x_u(t;\theta^w) \\
&= s(x_u(t;\theta^w))c(s(x_u(t;\theta^w)))\frac{\partial}{\partial \theta^w}x_u(t;\theta^w) \\
&= \delta_u(t;\theta^w)c(\delta_u(t;\theta^w))\frac{\partial}{\partial \theta^w}x_u(t;\theta^w).
\end{aligned}$$

Inserting this into the above given derivative of $w_u(t;\theta^w)$, we obtain

$$\begin{aligned}
\frac{\partial}{\partial \theta^w} w_u(t;\theta^w) &= \sum_{i=0}^{n-1} (-1)^{q_{u_i}(u_{i+1})} \frac{\partial}{\partial \theta^w}\delta_{u_i}(t;\theta^w) \cdot \prod_{\substack{j=0 \\ j\neq i}}^{n-1} c^{q_{u_j}(u_{j+1})}(\delta_{u_j}(t;\theta^w)) \\
&= \sum_{i=0}^{n-1} (-1)^{q_{u_i}(u_{i+1})} \delta_{u_i}(t;\theta^w) c(\delta_{u_i}(t;\theta^w)) \frac{\partial}{\partial \theta^w} x_{u_i}(t;\theta^w) \\
&\qquad \cdot \prod_{\substack{j=0 \\ j\neq i}}^{n-1} c^{q_{u_j}(u_{j+1})}(\delta_{u_j}(t;\theta^w)).
\end{aligned}$$

Here, we want to use the fact that

$$c(x)\cdot x = c^1(x)\cdot c^0(x) = c^q(x)\cdot c^{1-q}(x) \quad \text{for } q \in \{0,1\},$$

and can thus write

$$\delta_{u_i}(t;\theta^w)c(\delta_{u_i}(t;\theta^w)) = c^{q_{u_i}(u_{i+1})}(\delta_{u_i}(t;\theta^w))c^{1-q_{u_i}(u_{i+1})}(\delta_{u_i}(t;\theta^w)).$$

The term $c^{q_{u_i}(u_{i+1})}(\delta_{u_i}(t;\theta^w))$ is the term which is missing in the last product of the above derivation; we can include it there, and the product becomes equal to $w_u(t;\theta^w)$. We get:

$$\begin{aligned}\frac{\partial}{\partial\theta^w}w_u(t;\theta^w) &= w_u(t;\theta^w)\cdot\sum_{i=0}^{n-1}(-1)^{q_{u_i}(u_{i+1})}c^{1-q_{u_i}(u_{i+1})}(\delta_{u_i}(t;\theta^w))\frac{\partial}{\partial\theta^w}x_{u_i}(t;\theta^w)\\ &= w_u(t;\theta^w)\cdot\left[\sum_{\substack{i=0\\ q_{u_i}(u_{i+1})=0}}^{n-1}(1-\delta_{u_i}(t;\theta^w))\frac{\partial}{\partial\theta^w}x_{u_i}(t;\theta^w)\right.\\ &\qquad\left.-\sum_{\substack{i=0\\ q_{u_i}(u_{i+1})=1}}^{n-1}\delta_{u_i}(t;\theta^w)\frac{\partial}{\partial\theta^w}x_{u_i}(t;\theta^w)\right]\\ &= w_u(t;\theta^w))\cdot\left[\sum_{\substack{i=0\\ q_{u_i}(u_{i+1})=0}}^{n-1}\frac{\partial}{\partial\theta^w}x_{u_i}(t;\theta^w)-\sum_{i=0}^{n-1}\delta_{u_i}(t;\theta^w)\frac{\partial}{\partial\theta^w}x_{u_i}(t;\theta^w)\right].\end{aligned}$$

Additionally, we find that

$$\frac{\partial x_u(t;\theta^w)}{\partial\alpha_u}=\alpha_u^\top\frac{\partial x(t)}{\partial\alpha_u}+x(t)\qquad\text{and}\qquad\frac{\partial x_u(t;\theta^w)}{\partial\beta_u}=\alpha_u^\top\frac{\partial x(t)}{\partial\beta_u}-1.$$

We see that in both cases the derivative of the weight functions is itself a function on the weight functions and the basis weight functions or the components of the decision maps, respectively. This can be effectively used in implementations.

Computation of the Levenberg-Marquardt descent direction Our presentation follows now Sjöberg et al. [1994]. The general formula for the update of the parameter vector θ in one optimization step is the following:

$$\theta^{\text{new}}:=\theta+\mu p,$$

where $\mu>0$ denotes the ***step size*** and the $(d\times 1)$ vector p denotes the ***descent direction***. We can generally take every $(d\times 1)$ vector p which fulfills

$$V_N'(\theta)p<0$$

as descent direction (V_N' being the derivative with respect to θ of the error function V_N). One usually chooses p to be of the form

$$p:=-R(\theta)^{-1}V_N'(\theta)^\top$$

with (symmetric) positive definite matrix $R(\theta)$. Then, also $R(\theta)^{-1}$ is positive definite, and

$$V_N'(\theta)p=-V_N'(\theta)R(\theta)^{-1}V_N'(\theta)^\top<0$$

holds. The easiest choice for $R(\theta)$ here is of course the $(d \times d)$ unity matrix

$$R(\theta) := I,$$

$p = -V_N'(\theta)^\top$ is then called ***gradient descent direction***. But the choice of this descent direction leads to a quite slow convergence of the procedure. In contrast, it is known that near a local minimum of the error function $V_N(\theta)$, the ***Newton descent direction***

$$p = -V_N''(\theta)^{-1} V_N'(\theta)^\top,$$

i.e.

$$R(\theta) := V_N''(\theta)$$

with

$$V_N''(\theta) := \frac{d^2}{d\theta^2} V_N(\theta),$$

results in an essentially better convergence speed. To be able to apply the Newton direction, the computation of the second derivative (Hesse matrix) $V_N''(\theta)$ is necessary, which in turn may lead to numerical problems. Nevertheless, we always require the positive definiteness of the matrix $R^{-1}(\theta)$ for every descent direction, which is not guaranteed for the choice $R(\theta) := V_N''(\theta)$ if θ is far away from the minimum. This is the reason why instead of $V_N''(\theta)$ one chooses the positive definite $(d \times d)$ matrix

$$H(\theta) := \frac{1}{N} \sum_{k=1}^{N} \Big(\frac{\partial}{\partial \theta} \Gamma(u(\cdot);\theta)(t_k) \Big)^\top \Big(\frac{\partial}{\partial \theta} \Gamma(u(\cdot);\theta)(t_k) \Big) = \frac{1}{N} \sum_{k=1}^{N} \psi_k^\top \psi_k$$

with the $(1 \times d)$ vectors $\psi_k := \frac{\partial}{\partial \theta} \Gamma(u(\cdot);\theta)(t_k)$ for $k = 1, \ldots, N$. This choice is guided by the decomposition

$$\begin{aligned} V_N''(\theta) &= \frac{1}{N} \sum_{k=1}^{N} \Big(\frac{\partial}{\partial \theta} \Gamma(u(\cdot);\theta)(t_k) \Big)^\top \Big(\frac{\partial}{\partial \theta} \Gamma(u(\cdot);\theta)(t_k) \Big) \\ &\quad - \frac{1}{N} \sum_{k=1}^{N} \big(y_k - \Gamma(u(\cdot);\theta)(t_k) \big) \frac{\partial^2}{\partial \theta^2} \Gamma(u(\cdot);\theta)(t_k) \\ &= H(\theta) - \frac{1}{N} \sum_{k=1}^{N} \big(y_k - \Gamma(u(\cdot);\theta)(t_k) \big) \frac{\partial^2}{\partial \theta^2} \Gamma(u(\cdot);\theta)(t_k). \end{aligned}$$

The choice $R(\theta) := H(\theta)$ is called the ***Gauß-Newton descent direction***. But, if the matrix $H(\theta)$ is ill-conditioned, the inversion of $H(\theta)$ may lead to problems, which can be avoided by the combination of the gradient direction with the Gauß-Newton direction:

$$R(\theta) := H(\theta) + \delta I \qquad \text{for a } \delta > 0.$$

This is the so called ***Levenberg-Marquardt descent direction*** and is the one we choose for our optimization.

Remark: The Levenberg-Marquardt descent direction is actually the Tikhonov regularization of the linearization of the operator

$$\big(\Gamma(u(\cdot);.)(t_k)\big)_{k=1}^{N} : \Theta \longrightarrow \mathbb{R}^N$$

at the point $\theta \in \Theta$ (see Engl et al. [2000], S. 285).

The optimal choice of the ***regularization parameter*** δ is still an open problem. According to Engl et al. [2000] it shouldn't be chosen too small. It is possible to compute the necessary inversion of the matrix $R(\theta)$ directly or by means of the singular value decomposition of $R(\theta)$.

Computation of the Armijo step size After the decision for the gradient descent direction has been made, the question for the choice of the step size $\mu > 0$ arises. The ideal step size using the Newton descent direction or the corresponding approximations near a local minimum is $\mu = 1$. But if θ is too far away from this minimum, the step size can be too large; in this case the error V_N may even increase. To avoid this, it should be guaranteed that the inequality

$$V_N(\theta + \mu p) \leq V_N(\theta) + \alpha\mu V_N'(\theta)p$$

holds for a (fixed) $\alpha \in (0, \frac{1}{2})$, which for small $\mu > 0$ always can be fulfilled (see e.g. Werner [1992b], p. 165ff). The choice of α inside the above interval guarantees the superlinear convergence of the quasi-Newton procedure. The ***Armijo step size*** can be found as follows: Beginning with the step size $\mu_0 = 1$ the above inequality will be tested and the step size iteratively decreased until the inequality is fulfilled. Here, one chooses

$$\mu_{s+1} \in [l\mu_s, u\mu_s] \quad \text{with } 0 < l \leq u < 1 \text{ and } s = 0, 1, \ldots.$$

If one chooses $l := u := \rho$ for a $\rho \in (0,1)$, one gets $\mu = \rho^s$, where s is the smallest non-negative integer with

$$V_N(\theta + \rho^s p) \leq V_N(\theta) + \alpha\rho^s V_N'(\theta)p.$$

We may set e.g. $\alpha := 1/4$ and $\rho := 1/2$. Remark that (at least) after some optimization steps the step size obtains the optimal value $\mu = 1$.

Complying with constraints of the parameters Some parameters have to follow constraints, for example scaling parameters σ need to be positive. After an update of the parameter vector θ by means of

$$\theta^{\text{new}} := \theta + \mu p$$

these constraints for the components of θ^{new} may be broken. To ensure complying with the constraints, we use the following method: We think each parameter θ_j, $j = 1, \ldots, d$, which shall be constraint to the open set $I_j \subset \mathbb{R}$, as an image of a bijective differentiable function

$$\zeta_j : \mathbb{R} \to I_j.$$

Thus $\theta_j = \zeta_j(\bar{\theta}_j)$ holds for a pre-image $\bar{\theta}_j := \zeta_j^{-1}(\theta_j)$. We get the map

$$\zeta : \mathbb{R}^d \longrightarrow I_1 \times \cdots \times I_d, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix} \longmapsto \zeta(x) := \begin{pmatrix} \zeta_1(x_1) \\ \vdots \\ \zeta_d(x_d) \end{pmatrix},$$

such that $\theta = \zeta(\bar{\theta})$ for $\theta := (\theta_1, \ldots, \theta_d)^\top$ and $\bar{\theta} := (\bar{\theta}_1, \ldots, \bar{\theta}_d)^\top = \zeta^{-1}(\theta)$ holds. The advantage of the newly introduced parameters $\bar{\theta}_j$ is that they do not need to obey any constraints. We may therefore update $\bar{\theta}$ instead of θ. After the application of ζ, the constraints with respect to the original parameters will be fulfilled automaticly. The update for $\bar{\theta}$ is

$$\bar{\theta}^{\text{new}} := \bar{\theta} + \bar{\mu}\bar{p},$$

where here the step size $\bar{\mu}$ and the descent direction $\bar{p}$ with respect to the transformed parameter vector $\bar{\theta}$ have to be computed in the same way as was described above for θ. The update-rule for the original parameter $\theta = \zeta(\bar{\theta})$ follows then to be:

$$\theta^{\text{new}} := \zeta(\bar{\theta}^{\text{new}}) = \zeta(\bar{\theta} + \bar{\mu}\bar{p}) = \zeta(\zeta^{-1}(\theta) + \bar{\mu}\bar{p}).$$

Considering the error function with respect to $\bar{\theta}$

$$\bar{V}_N(\bar{\theta}) := V_N \circ \zeta(\bar{\theta}),$$

the descent direction $\bar{p}$ is given by

$$\bar{p} := -\bar{R}(\bar{\theta})^{-1} \frac{\partial}{\partial \theta} \bar{V}_N(\bar{\theta})^\top = -\bar{R}(\bar{\theta})^{-1} \frac{\partial}{\partial \theta} V_N(\zeta(\bar{\theta}))^\top$$

with

$$\bar{R}(\bar{\theta}) := \bar{H}(\bar{\theta}) + \delta I, \qquad \delta > 0.$$

Here, we set

$$\begin{aligned} \bar{H}(\bar{\theta}) :=& \frac{1}{N} \sum_{k=1}^{N} \Big(\frac{\partial}{\partial \theta} \Gamma(u(\cdot); \zeta(\bar{\theta}))(t_k) \Big)^\top \Big(\frac{\partial}{\partial \theta} \Gamma(u(\cdot); \zeta(\bar{\theta}))(t_k) \Big) \\ =& \frac{1}{N} \sum_{k=1}^{N} \bar{\psi}_k(\bar{\theta})^\top \bar{\psi}_k(\bar{\theta}) \end{aligned}$$

with

$$\bar{\psi}_k(\bar{\theta}) := \frac{\partial}{\partial \theta} \Gamma(u(\cdot); \zeta(\bar{\theta}))(t_k).$$

It follows from the chain rule that the gradient of the error function $\bar{V}_N'(\bar{\theta}) := \partial \bar{V}_N(\bar{\theta}) / \partial \bar{\theta}$ is given by

$$\begin{aligned} \frac{\partial}{\partial \theta} \bar{V}_N(\bar{\theta}) = \frac{\partial}{\partial \theta} V_N(\zeta(\bar{\theta})) =& -\frac{1}{N} \sum_{k=1}^{N} \big| y_k - \Gamma(u(\cdot); \zeta(\bar{\theta}))(t_k) \big| \cdot \frac{\partial}{\partial \theta} \Gamma(u(\cdot); \zeta(\bar{\theta}))(t_k) \\ =& -\frac{1}{N} \sum_{k=1}^{N} \big| y_k - \Gamma(u(\cdot); \zeta(\bar{\theta}))(t_k) \big| \cdot \bar{\psi}_k(\bar{\theta}). \end{aligned}$$

All these equations lead to the conclusion that the computation of

$$\bar{\psi}_k(\bar{\theta}) = \frac{\partial}{\partial \theta}\Gamma(u(\cdot);\zeta(\bar{\theta}))(t_k)$$

is essential for the computation of the step size $\bar{p}$. We get again by the chain rule:

$$\bar{\psi}_k(\bar{\theta}) = \frac{\partial}{\partial \theta}\Gamma(u(\cdot);\zeta(\bar{\theta}))(t_k) = \Big(\frac{\partial}{\partial \zeta(\theta)}\Gamma(u(\cdot);\zeta(\bar{\theta}))(t_k)\Big)\Big(\frac{\partial}{\partial \theta}\zeta(\bar{\theta})\Big) = \psi_k(\theta)\frac{\partial}{\partial \theta}\zeta(\bar{\theta})$$

because

$$\psi_k(\theta) = \frac{\partial}{\partial \theta}\Gamma(u(\cdot);\theta)(t_k) = \frac{\partial}{\partial \zeta(\theta)}\Gamma(u(\cdot);\zeta(\bar{\theta}))(t_k).$$

Since the j-th component θ_j of θ depends only on the j-th component $\bar{\theta}_j$ of $\bar{\theta}$, the matrix $\partial\zeta(\bar{\theta})/\partial\bar{\theta} = \mathrm{diag}(\zeta_j'(\bar{\theta}_j))$ is diagonal with $\zeta_j'(\bar{\theta}_j) := \partial\zeta_j(\bar{\theta}_j)/\partial\bar{\theta}_j$. Considering the $(1\times d)$ vector

$$\bar{\psi}_k(\bar{\theta}) = (\bar{\psi}_{k,1}(\bar{\theta}),\ldots,\bar{\psi}_{k,d}(\bar{\theta}))$$

with components

$$\bar{\psi}_{k,j}(\bar{\theta}) := \frac{\partial}{\partial \theta_j}\Gamma(u(\cdot);\zeta(\bar{\theta}))(t_k)$$

for $j = 1,\ldots,d$, this yields

$$\bar{\psi}_{k,j}(\bar{\theta}) = \frac{\partial}{\partial \theta_j}\Gamma(u(\cdot);\zeta(\bar{\theta}))(t_k) = \frac{\partial}{\partial \theta_j}\Gamma(u(\cdot);\theta)(t_k)\frac{\partial}{\partial \theta_j}\zeta_j(\bar{\theta}_j) = \psi_{k,j}(\theta)\cdot\zeta_j'(\bar{\theta}_j)$$

where

$$\psi_{k,j}(\theta) := \frac{\partial}{\partial \theta_j}\Gamma(u(\cdot);\theta)(t_k).$$

The step size $\bar{\mu}$ is given by the Armijo step size with respect to the newly derived update-rules: One decreases the initial step size $\bar{\mu} = \bar{\mu}_0 := 1$ until the inequality

$$\bar{V}_N(\bar{\theta}+\bar{\mu}\bar{p}) \le \bar{V}_N(\bar{\theta}) + \alpha\bar{\mu}\bar{V}_N'(\bar{\theta})\bar{p},$$

i.e. until

$$V_N(\zeta(\zeta^{-1}(\theta)+\bar{\mu}\bar{p})) \le V_N(\theta) + \alpha\bar{\mu}\Big(\frac{\partial}{\partial\theta}V_N(\zeta(\bar{\theta}))\Big)\bar{p}$$

is fulfilled. In table 1.2 we show for the respective intervals $I_j = \mathbb{R}$, $I_j = (0,\infty)$ and $I_j = (0,1)$ some possible transformations ζ_j together with the corresponding maps ζ_j^{-1} and ζ_j' as well as the transformed gradients $\bar{\psi}_{k,j}(\bar{\theta})$ and the parameter updates θ_j^{new}. Note that in an implementation of the algorithm only the computation of the formulas in the last two rows of the table is necessary, i.e. it is enough to transform the gradients $\psi_{k,j}$ to the gradients $\bar{\psi}_{k,j}$ in the described way (the descent direction $\bar{p} = (\bar{p}_1,\ldots,\bar{p}_d)^\top$ and the step size $\bar{\mu}$ are then computed with these transformed gradients in the usual way) and to apply the parameter update as shown in the last row of the table. Obviously, the explicit computation of the transformed parameters $\bar{\theta}$ is not necessary.

	Constraints for the parameter θ_j		
	$\theta_j \in \mathbb{R}$	$\theta_j > 0$	$0 < \theta_j < 1$
I_j	$\mathbb{R}$	$(0,\infty)$	$(0,1)$
$\zeta_j(\bar{\theta}_j)$	$\mathrm{Id}(\bar{\theta}_j)$	$\exp(\bar{\theta}_j)$	$\frac{1}{1+\exp(-\theta_j)}$
$\zeta_j^{-1}(\theta_j)$	$\mathrm{Id}(\theta_j)$	$\ln(\theta_j)$	$\ln\frac{\theta_j}{1-\theta_j}$
$\zeta_j'(\bar{\theta}_j)$	1	$\zeta_j(\bar{\theta}_j)$	$\zeta_j(\bar{\theta}_j)(1-\zeta_j(\bar{\theta}_j))$
$\bar{\psi}_{k,j}(\bar{\theta})$	$\psi_{k,j}(\theta)$	$\psi_{k,j}(\theta)\cdot\theta_j$	$\psi_{k,j}(\theta)\cdot\theta_j\cdot(1-\theta_j)$
θ_j^{new}	$\theta_j+\bar{\mu}\bar{p}_j$	$\theta_j\cdot\exp(\bar{\mu}\bar{p}_j)$	$\frac{\theta_j}{\theta_j+(1-\theta_j)\exp(-\bar{\mu}\bar{p}_j)}$

Table 1.2: Parameter transformations ensuring complying with the constraints

1.4.3 Applications of the gradient based optimization to the improvement of the LOLIMOT algorithm

The gradient based optimization will enable the following improvements of the LOLIMOT algorithm:

- More flexibility for the division of the regime space. The restriction that the regime space is only divided by axis parallel partitions can be given up. Both the decision tree based weight functions as well as the optimization algorithm allow translations and rotations of the dividing hyperplanes. This should lead to a global model built by less local models.

- The translation and rotation of the dividing hyperplanes may lead to the hiding of some regimes, i.e. the resulting weight function can be near zero everywhere, such that these regimes loose their influence on the global model. These regimes can be deleted without changing the global model too much. The decision tree is pruned in this way. This also leads to smaller global models.

- The parameters initially estimated for ARX models can be adapted to OE models, which are better suited for simulation.

These points will be explained further in the next three paragraphs.

Generalizing the dividing hyperplanes

The dividing hyperplane in the original decision tree $T = (G, r)$, $G = (V, E)$, at a leaf $u \in \mathscr{L}_T$ is given by the equation

$$\psi_i = \beta_u \qquad \text{for some given } i \in 1, \ldots, m \text{ and } \beta_u \in \mathbb{R}$$

where $\psi = (\psi_1, \ldots, \psi_m)^\top \in \mathbb{R}^m$ is an element of the regime space $\Psi := h^w(x)$. If we define

$$\alpha_u := e_i$$

where $e_i \in \mathbb{R}^m$ is the unit vector whose i-th coefficient is one and the other coefficients are zero, we can write this in the equivalent form

$$\alpha_u^\top \psi = \beta_u.$$

If we allow α_u to be any vector in $\mathbb{R}^m$, this results in hyperplanes which are not restricted to be parallel to one of the axes.

The optimization proceeds as follows:

- Take $\alpha_u^{(0)}$ obtained by the LOLIMOT algorithm as initial value.
- Use the Levenberg-Marquardt gradient based update to get $\alpha_u^{(n+1)}$ from $\alpha_u^{(n)}$; use the whole vector $\alpha_u^{(n)}$ for the optimization.

Pruning the decision tree

The translation and rotation of the dividing hyperplanes obtained in the foregoing subsection can result in a hiding of certain regimes, as can be seen in figure 1.13. These regimes can be

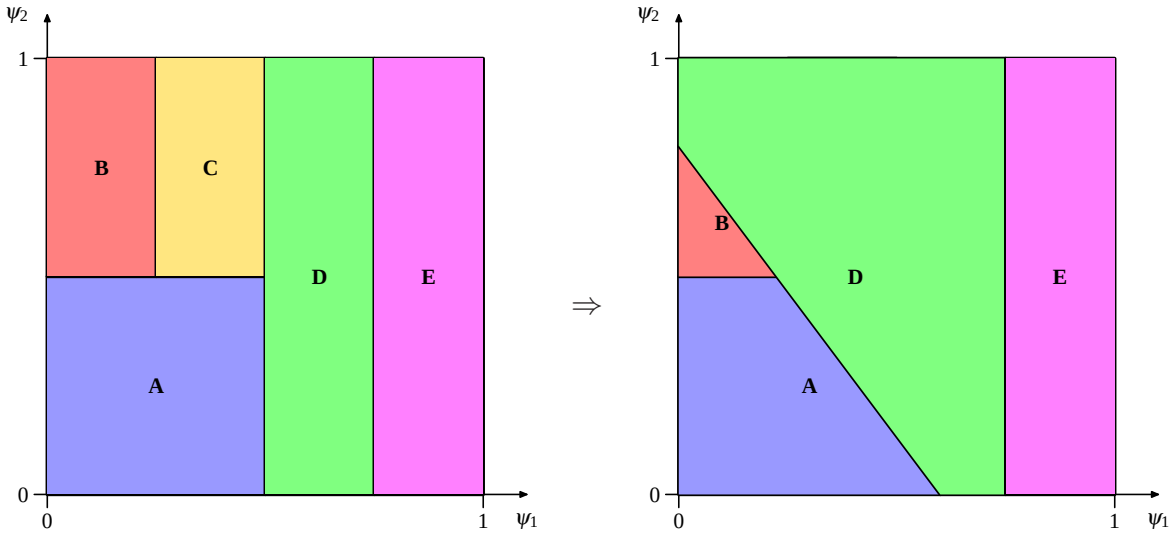


Figure 1.13: Hidden regime C after rotation of a hyperplane

deleted. If the splits are strict, the global model does not change after pruning. If the splits are smooth, the global model will change only minimally. We have to decide algorithmically which regimes are hidden. We first provide the terminology from analytical geometry.

Definition 1.14: *Let $d \in \mathbb{N}$.*

- *A **hyperplane** H in $\mathbb{R}^d$ is given by the solution of an affine-linear equation, i.e.*

$$H := \left\{ \omega \in \mathbb{R}^d \;\middle|\; \alpha^\top \omega = \beta \right\}$$

  for some $\alpha \in \mathbb{R}^d$, $\beta \in \mathbb{R}$.

- *An **(open) half space** A is given by the solution of an affine-linear inequality, i.e.*

$$A := \left\{ \omega \in \mathbb{R}^d \;\middle|\; \alpha^\top \omega > \beta \right\}$$

  for some $\alpha \in \mathbb{R}^d$, $\beta \in \mathbb{R}$.

- *A **convex (open) polytope** Π is given by the intersection of finitely many (open) half spaces, i.e.*

$$\Pi := \bigcap_{i=1}^{n} A_i$$

  for half spaces A_i, $i = 1, \ldots, n$.

A convex polytope is indeed convex as a set in $\mathbb{R}^d$. Therefore, a hyperplane H, say given by $\alpha^\top \omega = \beta$, divides the space $\mathbb{R}^d$ into three disjoint parts: The hyperplane itself and two open half spaces

$$A_0 := \left\{ \omega \in \mathbb{R}^d \;\middle|\; -\alpha^\top \omega > -\beta \right\}$$

and

$$A_1 := \left\{ \omega \in \mathbb{R}^d \;\middle|\; \alpha^\top \omega > \beta \right\}.$$

If we further consider an open convex polytope Π and the intersection of Π with the hyperplane H, we have only two possibilities:

- Either the intersection $\Pi \cap H$ is empty; then the polytope Π remains untouched.
- Or the intersection $\Pi \cap H$ is not empty, and then we have a division of Π into three disjoint nonempty convex parts: the two convex open polytopes $\Pi_0 := \Pi \cap A_0$ and $\Pi_1 := \Pi \cap A_1$, and the "slice" $\Pi \cap H$ (being an open convex polytope in the dimension $d-1$).

Neglecting the slice $\Pi \cap H$, further hyperplanes divide the resulting polytopes into new polytopes (and the remaining slices). We can thus define:

Definition 1.15: *Let $T = (G, r)$, $G = (V, E)$, be a full binary decision tree with splitting rules $\alpha_u \omega > \beta_u$ associated to each of its inner vertices $u \in \mathscr{N}_T$.*

- *To each inner vertex $u \in \mathscr{N}_T$, the **associated hyperplane** H_u is given by*

$$H_u := \left\{ \omega \in \mathbb{R}^d \;\middle|\; \alpha_u \omega = \beta_u \right\}.$$

- *To each edge $(u,v) \in E$ with $u > v$, the **associated half space** $A_{(u,v)}$ is given by*

$$A_{(u,v)} := \left\{ \omega \in \mathbb{R}^d \;\middle|\; \zeta_{(u,v)} \alpha_u \omega > \zeta_{(u,v)} \beta_u \right\}$$

where

$$\zeta_{(u,v)} := (-1)^{1-q_u(v)} = \begin{cases} -1 & \text{if } q_u(v) = 0, \\ +1 & \text{if } q_u(v) = 1. \end{cases}$$

- *To each vertex $u \in V$, let $(u_0, \ldots, u_n)$, $u_0 = r$, $u_n = u$ be the path from the root r to u. Then the **associated polytope** Π_u is given by*

$$\Pi_u := \bigcap_{i=0}^{n-1} A_{(u_i,u_{i+1})},$$

with the convention $\Pi_r := \mathbb{R}^d$.

Definition 1.16: *Let $T = (G,r)$, $G = (V,E)$, be a full binary decision tree with the decision map given by splitting rules, let $u \in V$ be a vertex of T and Π_u be the associated polytope. We call the vertex u **hidden** if $\Pi_u = \emptyset$.*

Lemma 1.2: *Let $T = (G,r)$, $G = (V,E)$, be a full binary decision tree with the decision map given by splitting rules. Let $u \in \mathcal{N}_T$ be an inner vertex and u_0, u_1 its children. Then: u is hidden if and only if both its children u_0 and u_1 are hidden. Especially, if u is hidden, then both its children u_0 and u_1 are hidden, and thus all its descendants are also hidden.*

Proof. From the definition of the associated polytopes it follows that

$$\Pi_u = \Pi_{u_0} \dot\cup \Pi_{u_1} \dot\cup (H_u \cap \Pi_u).$$

That u_0 and u_1 are hidden means that the associated polytopes Π_{u_0} and Π_{u_1} are both empty:

$$\Pi_{u_0} = \Pi_{u_1} = \emptyset.$$

If u is hidden, i.e. $\Pi_u = \emptyset$, then this follows immediately.

If, on the other hand, this is given, we get

$$\Pi_u = (H_u \cap \Pi_u).$$

As finite intersection of open sets, Π_u is itself open. H_u as a hyperplane is a null-set (the Lebesgue measure is 0), and so is $\Pi_u = (H_u \cap \Pi_u)$. As a null-set, Π_u contains no open ball of positive radius, and can thus be open only in the case $\Pi_u = \emptyset$. We have thus proved that u is hidden. □

To check if a given vertex is hidden, one has to check whether the associated polytope

$$\Pi_u = \bigcap_{i=0}^{n-1} A_{u_i,u_{i+1}},$$

$(u_0,\dots,u_n)$ being a path from r to u, is empty. This in turn results in showing that the inequalities

$$\zeta_{(u_i,u_{i+1})}\alpha_{u_i}\omega > \zeta_{(u_i,u_{i+1})}\beta_{u_i}, \qquad i=0,\dots,n-1,$$

are not simultaneously solvable.

We thus have the following algorithm to detect the hidden vertices:

- Begin with the root r. Since $\Pi_r = \mathbb{R}^d$, the root is never hidden. Since $\Pi_r = \mathbb{R}^d$ has a non-empty intersection with every non-empty hyperplane H_r, also the children of the root are not hidden. Thus: proceed by recursively testing the grandchildren of the root.
- Test of a vertex u with a parent v which is not hidden:
  - If the "sibling" of u has already proved to be hidden, then u cannot be hidden (according to the lemma); recursively check its children.
  - Else, check if Π_u is empty in the following way: Let $(u_0,\dots,u_n)$, $u_0 = r$, $u_{n-1} = v$, $u_n = u$, be the path from the root r to u. If the inequalities

    $$\zeta_{(u_i,u_{i+1})}\alpha_{u_i}\omega > \zeta_{(u_i,u_{i+1})}\beta_{u_i}, \qquad i=0,\dots,n-1,$$

    are not simultaneously solvable then Π_u is empty (if there is a solution, this solution belongs to Π_u). Since $v = u_{n-1}$ is not hidden, we at least know that the inequalities up to $n-2$, i.e.

    $$\zeta_{(u_i,u_{i+1})}\alpha_{u_i}\omega > \zeta_{(u_i,u_{i+1})}\beta_{u_i}, \qquad i=0,\dots,n-2,$$

    are simultaneously solvable. If u is hidden, then all its descendants are also hidden. If u is not hidden, recursively check its children.

When we know which vertices are hidden and which not, we can easily prune them. Let u be a vertex which is not hidden, and let u_1 and u_2 be its children, u_2 hidden. Then u_1 is not hidden (according to the lemma); also, u cannot be the root, and has a parent v. The pruning of u_2 is done as follows: Delete the vertices u, u_2, and all descendants of u_2 together with all edges where one of the deleted vertices is incident in. Include the new edge $v \sim u_1$ into the edges. (The splitting rules remain unchanged, as far as the corresponding vertices have not been deleted.)

The algorithm may be to complex for using it in every iteration of the LOLIMOT algorithm. Possibly one uses it only once at the end.

Adapting the parameters to simulation purposes

The difference between NARX models and NOE models is in a different state vector: The NARX model has

$$x(t) = \big(u(t),u(t-1),\dots,u(t-n_u),y(t-1),\dots,y(t-n_y)\big)^\top$$

whereas the NOE model has

$$x(t) = \big(u(t),u(t-1),\dots,u(t-n_u),\hat{y}(t-1),\dots,\hat{y}(t-n_y)\big)^\top.$$

For simulation purposes the latter model representation, i.e. the NOE model, may be more adequate because during simulation only the computed outputs $\hat{y}$ are available. But the NOE model parameters are more difficult to compute because the outputs $\hat{y}$, which are necessary to estimate the parameters correctly, already depend on these parameters. Therefore this estimation is only possible with iterative methods, called pseudo-linear regression (see e.g. Ljung [1999]). In contrast, the estimation of the parameters of the NARX model is easily done by linear regression.

One could proceed in the following way: First construct a local model network of NARX type using the original LOLIMOT algorithm based upon weighted least squares regression. Additionally use one of the gradient-based improvements of the last paragraphs (oblique boundaries, pruning). But instead of using the non-recursive derivatives of the NARX model, one could as well use the recursive derivatives of the NOE model. Even if one does not want to use oblique boundaries or pruning, some iterations with the NOE model derivatives are possible, thus transforming and adjusting the local model network to a global NOE model.

2 Dealing with time: Dynamics

We already mentioned the difference between static and dynamical systems: While static systems give the same output for the same input at all times, the output of a dynamical system depends on the history of the system. Previous inputs influence the actual output equally well as the actual input does. Dynamical systems have been studied for several decades, especially in the cases of linear models with normal (Gaussian) disturbances. The theory for these normal linear models is well-developed and established. In the last years, interest has shifted more and more away from this special kind of models: From linearity to nonlinearity as well as from normality to non-normality. The increase in difficulty for theoretical and computational methods is tremendous. Linearization methods based on differentiability of nonlinear models are in use for a long time. But these methods often have to be considered only as approximations, and seldom a rigorous theoretical framework could have been established. Furthermore, linearization rises several problems: The differentiability of the systems has to be required; nevertheless, apart from necessary smoothness considerations, there exist dynamical systems which loose there typical behaviour when linearized. These systems are sometimes said to show hard nonlinearities. A typical example are systems with hysteresis. All this concerns linearity. Other problems occur when disturbances and noise are introduced into the systems. If this noise is Gaussian and is propagated by a linear system, the propagated noise is still Gaussian. If the system is nonlinear, the situation is completely different. The noise looses its Gaussianity and thus its analytical tractability.

To make things more apparent we first neglect disturbances and noise. We assume that our systems are deterministic: Equal excitations of a deterministic system with equal initial conditions lead always to an equal behaviour of the system. Non-determinism, which is the same as the introduction of uncertainties in our system effected e.g. by noise and disturbances, will be considered in the next chapter.

Overview The first section of this chapter is devoted to an axiomatic representation of a very wide range of deterministic dynamical systems. Within this background, we consider special kinds of dynamical systems like linear and differentiable systems, and hysteretic systems as an important example of systems with hard nonlinearities.

In the second section, we focus exclusively on those systems which exhibit a so-called rate independent hysteresis, especially Preisach hysteresis. The Preisach hysteresis is defined as a continuous superposition of simpler building blocks. This construction shows a quite general procedure, called atomic decomposition, and we will show that also the local model networks may be extended in this direction. We explore the important rôle of so-called reduced memory sequences for rate independent hysteresis. Reduced memory sequences contain the whole information about the present state of a hysteretic system. We describe some kind of

primitive function for Preisach hysteresis operators which can be used for both implementation and identification of those systems. For identification purposes, we develop a variant of the LOLIMOT algorithm.

Contributions

- The interpretation of local model networks as atomic decompositions and a generalization of local model networks derived from this.
- Slightly modified version of reduced memory sequences (called prefixed reduced memory sequences).
- Slightly generalized Preisach hysteresis und simpler version of summation formula.
- The identification of Preisach hysteresis by a variant of the LOLIMOT algorithm.

2.1 Deterministic models for dynamical systems

Our aim in the first part of the present chapter is to provide a frame for the treatment of deterministic nonlinear systems. Often, for reasons of simplicity and tractability, only linear or at least differentiable systems are considered. Linear systems are always defined on a (real or complex) vector space. Nonlinear differentiable systems are defined on a manifold and usually treated by linearization (see e.g. Banks [1988]). An important subclass are bilinear systems. But there are also other nonlinearities occurring in technical systems which cannot be linearized. These nonlinearities are therefore called hard nonlinearities, examples being

- systems with hysteresis,
- systems with discontinuities,
- systems with dead zones,
- etc.

Before we present a general axiomatic approach, we introduce the main terms we will use to describe dynamical systems by considering the most common dynamical systems, the linear differential systems.

A motivating example: Linear differential systems Consider the following classical "linear differential system":

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned}$$

where: A, B, C, D are suitable matrix functions and:

- $t \in \mathbb{R}$ is the time,
- $u(t)$ is the input at time t (given),
- $y(t)$ is the output at time t (observed),
- $x(t)$ is the state at time t (usually not observable),

and all values are taken from some finite-dimensional Euclidean space. Then the theory of linear differential equations tells us that, under certain conditions on A, B, C, D, we have for each initial value x_0 at initial time t_0 and for each input function u a unique solution $x(t)$. The state $x(t)$ accumulates the whole internal knowledge of the system at time t, whereas the output $y(t)$ depends directly on $x(t)$ and $u(t)$ at time t. Knowing the value of x at a given time t, we are able to exactly forecast the behaviour of the system in all future times $\tau > t$ (for each given input u).

How to generalize this?

Definition of deterministic state space systems

In the following presentation we provide the broad axiomatic definition for deterministic state space systems given in Hinrichsen and Pritchard [2005].

A ***dynamical system*** or ***deterministic state space system*** is defined as a 7-tuple

$$\Sigma = (\mathscr{T}, \mathscr{U}, \mathscr{U}^*, \mathscr{X}, \mathscr{Y}, \varphi, \eta)$$

where we further have to specify the data given in this tuple and the axioms reigning on these data.

Data We begin with the data:

- $\emptyset \neq \mathscr{T} \subseteq \mathbb{R}$ ***time domain*** $\rightarrow$ $t \in \mathscr{T}$ time
- $\emptyset \neq \mathscr{U}$ ***input value space*** $\rightarrow$ $u \in \mathscr{U}$ input value
- $\emptyset \neq \mathscr{U}^* \subseteq \mathscr{U}^{\mathscr{T}}$ ***input function space*** $\rightarrow$ $u(\cdot) \in \mathscr{U}^*$ input function
- $\emptyset \neq \mathscr{X}$ ***state space*** $\rightarrow$ $x \in \mathscr{X}$ state
- $\emptyset \neq \mathscr{Y}$ ***output value space*** $\rightarrow$ $y \in \mathscr{Y}$ output value
- $\varphi : \mathscr{D}_\varphi \longrightarrow \mathscr{X}$ ***state transition map*** $\rightarrow$ $x(t) = \varphi(t; t_0, x_0, u(\cdot))$
- $\eta : \mathscr{T} \times \mathscr{X} \times \mathscr{U} \longrightarrow \mathscr{Y}$ ***output map*** $\rightarrow$ $y(t) = \eta(t, x, u)$

with $\mathscr{D}_\varphi \subseteq \mathscr{T}^2 \times \mathscr{X} \times \mathscr{U}^*$.

Further terminology Before we write down the axioms, we need some further terminology. With the inputs into the system Σ given by

- $t_0 \in \mathscr{T}$ ***initial time***,
- $x_0 \in \mathscr{X}$ ***initial state***,
- $u(\cdot) \in \mathscr{U}^*$ ***input function***,

we define the ***life span*** by

$$\mathscr{T}_{t_0,x_0,u(\cdot)} := \left\{ t \in \mathscr{T} \;\middle|\; (t;t_0,x_0,u(\cdot)) \in \mathscr{D}_\varphi \right\}$$

and the ***state trajectory*** by

$$\varphi(\,.\,;t_0,x_0,u(\cdot)) : \mathscr{T}_{t_0,x_0,u(\cdot)} \longrightarrow \mathscr{X}$$

(see figure 2.1). For all $t \in \mathscr{T}_{t_0,x_0,u(\cdot)}$ we call

$$x(t) := \varphi(t;t_0,x_0,u(\cdot))$$

the ***state*** of Σ at time t and

$$y(t) := \eta(t,x(t),u(t))$$

the ***output*** of Σ at time t.

By an ***interval*** in $\mathscr{T} \subseteq \mathbb{R}$, we mean one of the following sets:

$$(a,b)\cap\mathscr{T}, \qquad (a,b]\cap\mathscr{T}, \qquad [a,b)\cap\mathscr{T}, \qquad [a,b]\cap\mathscr{T},$$

with $-\infty \le a \le b \le \infty$.

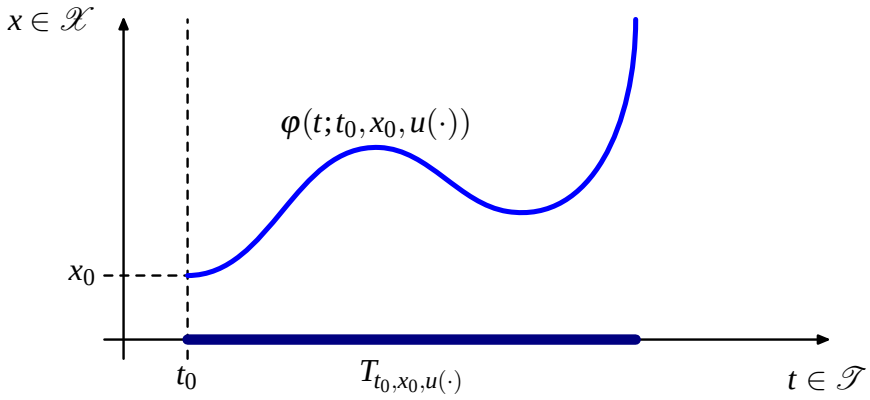


Figure 2.1: Life span and state trajectory

Axioms For Σ with the above given data to be a deterministic state space system we want the following four axioms to hold (Hinrichsen and Pritchard [2005]):

- ***Interval axiom***: If $t_0 \in \mathscr{T}$, $x_0 \in \mathscr{X}$, $u(\cdot) \in \mathscr{U}^*$ then the life span of $\varphi(\,.\,;t_0,x_0,u(\cdot))$, i.e.

  $$\mathscr{T}_{t_0,x_0,u(\cdot)} := \left\{ t \in \mathscr{T} \;\middle|\; (t;t_0,x_0,u(\cdot)) \in \mathscr{D}_\varphi \right\},$$

  is an interval in $\mathscr{T}$ containing t_0.

- ***Consistency axiom***: If $t_0 \in \mathscr{T}$, $x_0 \in \mathscr{X}$, $u(\cdot) \in \mathscr{U}^*$ then

$$\varphi(t_0; t_0, x_0, u(\cdot)) = x_0.$$

- ***Causality axiom***: If $t_0 \in \mathscr{T}$, $x_0 \in \mathscr{X}$, $u(\cdot), v(\cdot) \in \mathscr{U}^*$, $t_1 \in \mathscr{T}_{t_0,x_0,u(\cdot)} \cap \mathscr{T}_{t_0,x_0,v(\cdot)}$ and

$$u(t) = v(t) \qquad \text{for all } t \in [t_0, t_1)$$

  then

$$\varphi(t_1; t_0, x_0, u(\cdot)) = \varphi(t_1; t_0, x_0, v(\cdot)).$$

- ***Cocycle property***: If $t_0 \in \mathscr{T}$, $x_0 \in \mathscr{X}$, $u \in \mathscr{U}^*$, $t_1 \in \mathscr{T}_{t_0,x_0,u(\cdot)}$ and

$$x_1 := \varphi(t_1; t_0, x_0, u(\cdot)),$$

  then $\mathscr{T}_{t_1,x_1,u(\cdot)} \subseteq \mathscr{T}_{t_0,x_0,u(\cdot)}$ and

$$\varphi(t; t_0, x_0, u(\cdot)) = \varphi(t; t_1, x_1, u(\cdot))$$

  for all $t \in \mathscr{T}_{t_1,x_1,u(\cdot)}$.

Discussion of axioms We still follow Hinrichsen and Pritchard [2005]. The interval axiom allows the life span $\mathscr{T}_{t_0,x_0,u(\cdot)}$ of a state trajectory $\varphi(\,.\,; t_0, x_0, u(\cdot))$ for all $t_0 \in \mathscr{T}$, $x_0 \in \mathscr{X}$, $u(\cdot) \in \mathscr{U}^*$ to be shorter than the whole time domain $\mathscr{T}$, but it ensures that it is always an interval. The state trajectory is in this sense „connected“. The consistency guarantees that the initial state x_0 is really the initial state of the trajectory $\varphi(\,.\,; t_0, x_0, u(\cdot))$ at the initial time t_0. The causality axiom ensures both causality and determinism: Given initial time and initial state, equal inputs *before* a time t cause equal effects *at* time t. Thus, if two inputs $u(t_1)$ and $v(t_1)$ are equal or not at a time $t_1 \geq t$ (i.e. present or future inputs) has no influence on the state of the system at time t, whereas the output $y(t)$ directly can be influenced by $u(t)$ and $v(t)$ (i.e. the present input), respectively. This is the causality property: only the past does influence the system at a time t. Determinism says then additionally that the same influences cause the same effects. The cocycle property guarantees that the state $x(t)$ at time t contains the complete information on the system. If we started the system newly with initial state $x_0 = x(t)$, then the system would show the same behaviour as if we did not restart it. We might think of the state as accumulating the whole history of the system. It can be seen as the internal memory of the system.

Complete and reversible systems The interval axiom says that for given t_0, x_0 and $u(\cdot)$, the initial time t_0 is always contained in the life span, i.e. $t_0 \in \mathscr{T}_{t_0,x_0,u(\cdot)}$, or, written differently,

$$D_\varphi \supseteq \mathrm{Diag}(\mathscr{T}^2) \times \mathscr{X} \times \mathscr{U}^*.$$

The life span $\mathscr{T}_{t_0,x_0,u(\cdot)}$ tells us at which times t the state $x(t)$ (and so the output $y(t)$) is defined. We will consider some special cases (Hinrichsen and Pritchard [2005]).

Definition 2.1: *Let $\Sigma = (\mathscr{T}, \mathscr{U}, \mathscr{U}^*, \mathscr{X}, \mathscr{Y}, \varphi, \eta)$ be a deterministic state space system.*

(a) *Σ is called **complete** if for all $t_0 \in \mathscr{T}$, $x_0 \in \mathscr{X}$, $u(\cdot) \in \mathscr{U}^*$:*

$$\mathscr{T}_{t_0,x_0,u(\cdot)} \supseteq \mathscr{T}_{t_0} := \left\{ t \in \mathscr{T} \;\middle|\; t \geq t_0 \right\},$$

i.e.

$$D_\varphi \supseteq \mathscr{T}^2_{\geq} \times \mathscr{X} \times \mathscr{U}^*$$

with

$$\mathscr{T}^2_{\geq} := \left\{ (t, t_0) \in \mathscr{T}^2 \;\middle|\; t \geq t_0 \right\}.$$

(b) *Σ is called **reversible** if for all $t_0 \in \mathscr{T}$, $x_0 \in \mathscr{X}$, $u(\cdot) \in \mathscr{U}^*$:*

$$\mathscr{T}_{t_0,x_0,u(\cdot)} = \mathscr{T},$$

i.e.

$$D_\varphi = \mathscr{T}^2 \times \mathscr{X} \times \mathscr{U}^*.$$

Input-ouput operator The completeness of a system allows the definition of an input-output operator; this opens the door to functional analysis. From the causality axiom follows that, if $u(\cdot)\big|_{\mathscr{T}_{t_0}} = v(\cdot)\big|_{\mathscr{T}_{t_0}}$ for $u(\cdot), v(\cdot) \in \mathscr{U}^*$, then

$$y(\,.\,; t_0, x_0, u(\cdot))\big|_{\mathscr{T}_{t_0}} = y(\,.\,; t_0, x_0, v(\cdot))\big|_{\mathscr{T}_{t_0}}$$

holds. The restriction of $y(\cdot)$ to $\mathscr{T}_{t_0}$ depends thus only on the restriction of $u(\cdot)$ to $\mathscr{T}_{t_0}$. Defining ${\mathscr{U}^*}_{t_0} := \left\{ u(\cdot)\big|_{\mathscr{T}_{t_0}} \;\middle|\; u(\cdot) \in \mathscr{U}^* \right\}$, we can formulate:

Definition 2.2: *Let the deterministic state space system $\Sigma = (\mathscr{T}, \mathscr{U}, \mathscr{U}^*, \mathscr{X}, \mathscr{Y}, \varphi, \eta)$ be complete, i.e. for all $t_0 \in \mathscr{T}, x_0 \in \mathscr{X}, u(\cdot) \in \mathscr{U}^*$:*

$$\mathscr{T}_{t_0,x_0,u(\cdot)} \supseteq \mathscr{T}_{t_0} := \left\{ t \in \mathscr{T} \;\middle|\; t \geq t_0 \right\}.$$

*Then the **input-output operator** for fixed t_0 and x_0 is defined by*

$$\Gamma_{t_0,x_0} : {\mathscr{U}^*}_{t_0} \longrightarrow \mathscr{Y}^{\mathscr{T}_{t_0}}, \qquad u(\cdot) \mapsto y(\cdot) = y(\,.\,; t_0, x_0, u(\cdot))\big|_{\mathscr{T}_{t_0}}.$$

Differential dynamical systems We provide some important examples (Hinrichsen and Pritchard [2005]).

Examples: (1) Automaton: A deterministic state space system

$$\Sigma = (\mathscr{T}, \mathscr{U}, \mathscr{U}^*, \mathscr{X}, \mathscr{Y}, \varphi, \eta)$$

is a (deterministic) ***automaton*** if

- $\mathscr{T} \subseteq \mathbb{Z}$,

- $\mathscr{U}$, $\mathscr{X}$, $\mathscr{Y}$ are non-empty sets,
- $\mathscr{U}^* \subseteq \mathscr{U}^{\mathscr{T}}$,
- $x(\cdot) = \varphi(.;t_0,x_0,u(\cdot))$ is given recursively by

$$\begin{aligned}\varphi(t_0+k+1;t_0,x_0,u(\cdot)) &= \psi(\varphi(t_0+k;t_0,x_0,u(\cdot)),u(t_0+k)), \qquad k \in \mathbb{N},\\ \varphi(t_0;t_0,x_0,u(\cdot)) &= x_0\end{aligned}$$

  for a function $\psi : \mathscr{X} \times \mathscr{U} \longrightarrow \mathscr{X}$,
- $\eta(t,x,u) = \eta(x,u)$.

The dynamics of the automaton is thus given by the equations

$$\begin{aligned}x(t+1) &= \psi(x(t),u(t)),\\ y(t) &= \eta(x(t),u(t))\end{aligned}$$

for all $t = t_0, t_0+1, t_0+2, \ldots$.

The automaton is exactly the deterministic counterpart of the stochastic state-space systems we will consider in chapter 3. It shows clearly what in the probabilistic context will be the Markov property: For a given time t, the next state $x(t+1)$ and the output $y(t)$ depend solely on the state $x(t)$ and the input $u(t)$.

(2) Differential dynamical system: A deterministic state space system

$$\Sigma = (\mathscr{T},\mathscr{U},\mathscr{U}^*,\mathscr{X},\mathscr{Y},\varphi,\eta)$$

is a ***differential dynamical system*** if for $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$ it holds that:

- $\mathscr{T} \subseteq \mathbb{R}$ is an open interval,
- $\mathscr{U} \subseteq \mathbb{K}^m$,
- $\mathscr{U}^*$ some suitable function space (see below),
- $\mathscr{X} \subseteq \mathbb{K}^n$ is open,
- $\mathscr{Y} \subseteq \mathbb{K}^d$,
- $x(\cdot) = \varphi(.;t_0,x_0,u(\cdot))$ is given as follows: There exists $f : \mathscr{T} \times \mathscr{X} \times \mathscr{U} \longrightarrow \mathbb{K}^n$ such that for all $t_0 \in \mathscr{T}$, $x_0 \in \mathscr{X}$, $u(\cdot) \in \mathscr{U}^*$ the initial value problem

$$\begin{aligned}\dot{x}(t) &= f(t,x(t),u(t)) \qquad \text{for } t \geq t_0,\ t \in \mathscr{T},\\ x(t_0) &= x_0,\end{aligned}$$

  has a unique solution on a maximal open interval I,
- $\eta : \mathscr{T} \times \mathscr{X} \times \mathscr{U} \longrightarrow \mathscr{Y}$ is continuous.

The input function set $\mathscr{U}^*$ usually depends on the application, implying different conditions on the solutions to be existent and unique. Thus, $\mathscr{U}^*$ may be taken e.g. as (some subspace of) the space of (piecewise) continuous functions or of the space $L^1_{\mathrm{loc}}(\mathscr{T},\mathbb{K}^m)$ of locally integrable functions (i.e. of Lebesgue-measureable functions $f:\mathscr{T}\longrightarrow\mathbb{K}^m$ such that $\int_a^b\|f(t)\|dt<\infty$ for all $a,b\in\mathscr{T}$ with $a<b$).

Differential systems are surely among the most important dynamical systems in physics and many other fields. Their importance is so high that often the word dynamical system is used as a synonym for differential systems. In control theory also the discrete counterparts, the difference systems (which are special cases of automata) play an important rôle.

Examples of deterministic state space systems in some sense opposed to the differential dynamical systems are the range invariant systems, e.g. systems with (rate independent) hysteresis. Range invariance is defined via time transformations.

Time transformations One possibility to distinguish between certain kinds of state space systems is the view at their behaviour when the underlying time is changed by a so-called time transformation. We will now give a definition which is more general than in Hinrichsen and Pritchard [2005].

Definition 2.3: *Let $\psi:\mathbb{R}\longrightarrow\mathbb{R}$ be a function. We call ψ a **time transformation** if it is monotonously increasing (order preserving) and bijective.*

Examples:

- $\psi_\tau(t):=t+\tau$ for some $\tau\in\mathbb{R}$ is called a ***time shift***.
- $\psi^\lambda(t):=\lambda t$ for some $\lambda\in\mathbb{R}_{>0}$ is called a ***time scaling***.

If for example a function $u:\mathscr{T}\longrightarrow\mathscr{U}$ is given and if $\psi:\mathbb{R}\longrightarrow\mathbb{R}$ is a time transformation, then the transformed function with respect to ψ is given by

$$\tilde{u}:\psi(\mathscr{T})\longrightarrow\mathscr{U},\qquad \tilde{u}(\psi(t))=u(t),$$

see figure 2.2.

What happens to a system Σ if it is subject to a time transformation ψ?

Definition 2.4: *Let $\Sigma=(\mathscr{T},\mathscr{U},\mathscr{U}^*,\mathscr{X},\mathscr{Y},\varphi,\eta)$ be a deterministic state space system and let $\psi:\mathbb{R}\longrightarrow\mathbb{R}$ be a time transformation. Σ is called **invariant with respect to the time transformation** ψ if for each $t_0\in\mathscr{T}$, $x_0\in\mathscr{X}$, $u(\cdot)\in\mathscr{U}^*$ with $\psi(\mathscr{T}_{t_0,x_0,u(\cdot)})\subseteq\mathscr{T}$, there exists $\tilde{u}\in\mathscr{U}^*$ with*

$$\tilde{u}(\psi(t))=u(t)\qquad \text{for all } t\in\mathscr{T}_{t_0,x_0,u(\cdot)}$$

such that:

$$\varphi(t;t_0,x_0,u(\cdot))=\varphi(\psi(t);\psi(t_0),x_0,\tilde{u}(\cdot))\qquad \text{for all } t\in\mathscr{T}_{t_0,x_0,u(\cdot)}$$

and

$$\eta(t,x,u)=\eta(x,u)\qquad \text{for all } t\in\mathscr{T},\ x\in\mathscr{X},\ u\in\mathscr{U}.$$

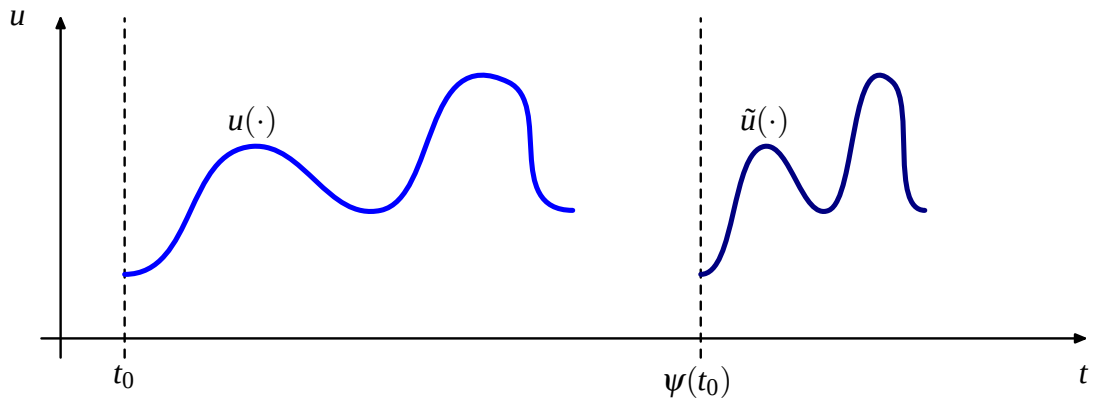


Figure 2.2: The function u and the transformed function $\tilde{u}$

In other words, a system Σ which is invariant with respect to a time transformation ψ does not recognize this transformation, in the sense that if the system produces an output y if an input u is given, then it produces a transformed output $\tilde{y} = y \circ \psi^{-1}$ if a transformed input $\tilde{u} = u \circ \psi^{-1}$ is provided. There are two important cases:

Definition 2.5: Σ *is called* ***time invariant*** *if it is invariant for every time shift*

$$\psi_\tau(t) := t + \tau \qquad \text{with } \tau \in \mathbb{R}.$$

This is often assumed. It means that if we start a system twice, at some time t_0 and at a time t_1, with the same (but time-shifted) input, then it will produce the same (time-shifted) output: the one beginning at time t_0 and the other beginning at time t_1.

The second important case is:

Definition 2.6: Σ *is called* ***range invariant*** *if it is invariant for* all *time transformations*

$$\psi : \mathbb{R} \longrightarrow \mathbb{R}.$$

This is a much stronger requirement than time invariance and serves as defining property of hysteresis (in the narrow sense of rate independent hysteresis). Indeed, range invariant systems and hysteretic systems are often considered as being equivalent; but the notion of hysteresis is actually broader, and range invariance (rate independence) should be seen as an extreme case of hysteresis. We return to this kind of hysteretic systems in full detail in the second part of this chapter.

Linear dynamical systems Another important property of dynamical systems is linearity (see again Hinrichsen and Pritchard [2005]).

Definition 2.7: *Let* $\Sigma = (\mathscr{T}, \mathscr{U}, \mathscr{U}^*, \mathscr{X}, \mathscr{Y}, \varphi, \eta)$ *be a deterministic state space system and let* $\mathbb{K}$ *be an arbitrary field.* Σ *is called* $\mathbb{K}$***-linear*** *if the following conditions hold:*

- $\mathscr{U}, \mathscr{U}^*, \mathscr{X}, \mathscr{Y}$ *are* $\mathbb{K}$*-vector spaces;*

- *for all* $t, t_0 \in \mathscr{T}$, $t \geq t_0$:
$$\varphi(t; t_0, \cdot, \cdot) : \mathscr{X} \times \mathscr{U}^* \longrightarrow \mathscr{X}$$
and
$$\eta(t; \cdot, \cdot) : \mathscr{X} \times \mathscr{U}^* \longrightarrow \mathscr{Y}$$
are $\mathbb{K}$*-linear maps.*

(Note that each of the maps is *jointly* linear in $\mathscr{X} \times \mathscr{U}^*$; this is not the same as bilinearity!) Every linear dynamical system is by definition complete and reversible. Thus, the input-output operator
$$\Gamma_{t_0,x_0} : \mathscr{U}^* \longrightarrow \mathscr{Y}^{\mathscr{T}}$$
exists for all $t_0 \in \mathscr{T}$, $x_0 \in \mathscr{X}$.

For linear systems the following important properties hold (Hinrichsen and Pritchard [2005]):

Lemma 2.1: *Let* $\Sigma = (\mathscr{T}, \mathscr{U}, \mathscr{U}^*, \mathscr{X}, \mathscr{Y}, \varphi, \eta)$ *be a* $\mathbb{K}$*-linear deterministic state space system for a field* $\mathbb{K}$*. Then the following holds:*

(a) Superposition principle*: For all* $t, t_0 \in \mathscr{T}$, $t \geq t_0$, $\lambda_i \in \mathbb{K}$, $x_{i,0}, x_i \in \mathscr{X}$, $u_i \in \mathscr{U}$, $u_i(\cdot) \in \mathscr{U}^*$, $i = 1, \ldots, k$*:*

$$\varphi\big(t; t_0, \sum_{i=1}^{k} \lambda_i x_{i,0}, \sum_{i=1}^{k} \lambda_i u_i(\cdot)\big) = \sum_{i=1}^{k} \lambda_i \varphi(t; t_0, x_{i,0}, u_i(\cdot)),$$
$$\eta\big(t; \sum_{i=1}^{k} \lambda_i x_i, \sum_{i=1}^{k} \lambda_i u_i\big) = \sum_{i=1}^{k} \lambda_i \eta(t; x_i, u_i).$$

(b) Decomposition principle*: For all* $t, t_0 \in \mathscr{T}$, $t \geq t_0$, $x_0 \in \mathscr{X}$, $u(\cdot) \in \mathscr{U}^*$*:*

$$\varphi(t; t_0, x_0, u(\cdot)) = \underbrace{\varphi(t; t_0, x_0, 0_{\mathscr{U}^*})}_{\textbf{\textit{free motion}}} + \underbrace{\varphi(t; t_0, 0_{\mathscr{X}}, u(\cdot))}_{\textbf{\textit{forced motion}}}.$$

Proof. The superposition principle is just the formulation of the $\mathbb{K}$-linearity. The decomposition into free and forced motion follows then immediately. □

We come back to our motivating example, the differential dynamical systems. We add linearity (Hinrichsen and Pritchard [2005]):

Example (Linear differential dynamical system)**:** A deterministic state space system $\Sigma = (\mathscr{T}, \mathscr{U}, \mathscr{U}^*, \mathscr{X}, \mathscr{Y}, \varphi, \eta)$ is a ***linear differential dynamical system*** if for $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$ the following holds:

- $\mathscr{T} \subseteq \mathbb{R}$ is an open interval,
- $\mathscr{U} = \mathbb{K}^m$,
- $\mathscr{U}^*$ some subspace of $L^1_{\mathrm{loc}}(\mathscr{T}, \mathbb{K}^m)$,

- $\mathscr{X} = \mathbb{K}^n$,
- $\mathscr{Y} = \mathbb{K}^d$,
- $x(\cdot) = \varphi(\,.\,;t_0,x_0,u(\cdot))$ is given as the *unique* solution of the linear initial value problem:

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \qquad \text{for all } t \in \mathscr{T}, \\ x(t_0) &= x_0, \end{aligned}$$

  and
- $\eta(t,x,u) = C(t)x + D(t)u$,

where A, B, C, D are suitable matrix functions (e.g. piecewise continuous).

The discrete-time counterpart, called ***linear difference dynamical system***, is given by the equations

$$\begin{aligned} x(t+1) &= A(t)x(t) + B(t)u(t) \qquad \text{for all } t \in \mathscr{T}, \\ y(t) &= C(t)x(t) + D(t)u(t). \end{aligned}$$

A linear differential or difference dynamical system is time-invariant if and only if the matrices $A(t)$, $B(t)$, $C(t)$, and $D(t)$ are constant with respect to the time t:

$$A(t) = A, \quad B(t) = B, \quad C(t) = C, \quad D(t) = D \quad \text{for all } t \in \mathscr{T}.$$

Such systems are called ***linear time-invariant*** (***LTI***) (differential or difference) dynamical systems and play a prominent rôle in systems and control theory.

Atomic decompositions At the end of this section we shortly mention a generalization of local model networks. By specialization it can be seen that also this kind of models fits into the given definition of deterministic state space systems.

For many operators Γ it is possible to find decompositions into simpler operators taken from a family

$$\{\Gamma^\omega \mid \omega \in \Omega\},$$

where the original operator Γ can be reconstructed from a continuous weighted superposition of the operators Γ^ω. Mathematically, this superposition is done by integration with respect to some measure μ_Γ, depending on the operator Γ and playing the rôle of the weights:

$$\Gamma = \int_{\omega\in\Omega} \Gamma^\omega d\mu_\Gamma(\omega).$$

The simpler operators Γ^ω used for the decomposition are called ***atoms***. The representation given in this way is thus called ***atomic decomposition*** of Γ. As mentioned and as we will present in more detail in the subsequent sections, Preisach hysteresis is defined following this general strategy. The atoms are in this case called ***hysterons***. But also local model networks can be seen as of this kind.

We need to be given:

- a measurable space $(\Omega, \mathfrak{A})$ (with a σ-algebra $\mathfrak{A}$ on Ω),
- a family of basis systems (atoms), $\Sigma_\omega = (\mathscr{T}, \mathscr{U}, \mathscr{U}^*, \mathscr{X}_\omega, \mathscr{Y}, \varphi_\omega, \eta_\omega)$, $\omega \in \Omega$,
- the joint state space $\mathscr{X} \subseteq \prod_{\omega\in\Omega} \mathscr{X}_\omega$, and
- a map $\mu : \mathfrak{A} \times \mathscr{T} \times \mathscr{X} \times \mathscr{U} \longrightarrow \mathbb{C}$.

Additionally we pose the following conditions on μ, φ_ω and η_ω: For each $t \in \mathscr{T}$, $x \in \mathscr{X}$, $u \in \mathscr{U}$:

- $\mu_{t,x,u} := \mu(\cdot; t, x, u) : \mathfrak{A} \longrightarrow \mathbb{C}$ is a (non-negative or signed or complex) measure,
- $\omega \mapsto \eta_\omega(t, x^\omega, u)$ is $\mu_{t,x,u}$-integrable.

For each $t, t_0 \in \mathscr{T}$, $x_0 \in \mathscr{X}$, $u(\cdot) \in \mathscr{U}^*$:

- $\Big(\varphi_\omega(t; t_0, x_0^\omega, u(\cdot))\Big)_{\omega\in\Omega} \in \mathscr{X}$.

Then the ***(generalized) local model network*** $\Sigma := (\mathscr{T}, \mathscr{U}, \mathscr{U}^*, \mathscr{X}, \mathscr{Y}, \varphi, \eta)$ is defined by:

$$\varphi(t; t_0, x_0, u(\cdot)) := \Big(\varphi_\omega(t; t_0, x_0^\omega, u(\cdot))\Big)_{\omega\in\Omega}, \qquad \eta(t, x, u) := \int_{\omega\in\Omega} \eta_\omega(t, x^\omega, u) d\mu_{t,x,u}.$$

This is a weighted continuous parallel superposition of the basis (or partial) systems Σ_ω, the weights given by the measures $\mu_{t,x,u}$. If the family Ω is finite and all measures $\mu_{t,x,u}$ are probability measures, i.e. they are non-negative and it holds

$$\int_{\omega\in\Omega} \mu_{t,x,u}(\omega) = 1,$$

then we are exactly in the case of Local Model Networks as presented in chapter 1.

2.2 Preisach hysteresis

In the previous section, we mentioned systems with hysteresis as an important example for systems with hard nonlinearities. Simple examples of hysteresis which occur naturally in technical systems are the phenomena of mechanical play and stop. More complex is the Preisach hysteresis, used for instance to model ferromagnetic phenomena.

Especially Preisach hysteresis shows a property completely new with respect to differential dynamical systems: ***long-time memory*** or ***nonlocal memory***. Long-time memory means the possibility of the system to internally store information about the history of the system which reaches an arbitrarily long time back into the past. Opposed to this is the ***short-time memory*** or ***local memory*** occurring in differential dynamical systems or their discrete counterparts, the difference dynamical systems. Short-time memory in this case is provided by the number of scalar states in the state-space representation. This number corresponds to the number of

derivatives in the continuous case or, in the discrete case, to the number of time steps the state signals can be tracked back into the past. These numbers are usually fixed and finite. In contrast, the long-time memory of a Preisach hysteresis cannot be stored in finite-dimensional vectors. The necessary information is better represented by a sequence of real values of finite or infinite length. In contrast to a vector, the length of this sequence is variable and even in the finite case principally unbounded. Taking the Preisach hysteresis as example, the information of the internal state of the corresponding system can be condensed into a sequence representing certain dominant minima and maxima of the input signal.

The investigation of hysteresis has developed into an interesting mathematical theory. In real systems, the differential and hysteresis properties mix and have to be modelled accordingly.

2.2.1 Definition and properties

Generally speaking, hysteresis is the "lagging of an effect behind its cause" (hysteresis comes from the Old Greek word $\acute{\upsilon}\sigma\tau\varepsilon\rho\varepsilon\tilde{\imath}\nu \equiv$ "to be late"). In physics, there are various definitions for hysteresis. In encyclopedias one finds more or less the following definition:

> Hysteresis in a dynamical system is a phenomenon wherein two (or more) time-dependent (physical) quantities bear a relationship which depends on the whole prior history (Walz [2000-2003]).

This notion is rather broad, and we will not follow it. There is another, more narrow definition of hysteresis which can be found in most mathematical books on hysteresis, for example in Visintin [1994]:

> Hysteresis is rate independent memory.

Thus, with scalar variables $u = u(t)$ and $y = y(t)$, the formation of hysteresis loops in the $(u(t), y(t))$-diagram is seen to be typical. If the hysteretic behaviour is modelled by a hysteresis operator H, i.e. $y = Hu$, the rate independence is given if

$$H(u \circ \psi^{-1}) = (Hu) \circ \psi^{-1}$$

for all time transformations ψ, i.e. if the hysteretical system is range invariant (see the previous section).

Examples of this kind of hysteresis are provided by the solutions of evolution variation inequalities; the stop operator for example is given by

$$\langle \dot{u}(t) - \dot{y}(t), y(t) - \tilde{y} \rangle \leq 0, \qquad \text{for all } \tilde{y} \in Z$$

where Z is a convex closed subset of a Hilbert space $\mathscr{H}$, where further $u : \mathscr{T} \longrightarrow \mathscr{H}$ is a given function, $y : \mathscr{T} \longrightarrow Z$ is an unknown function, both with time set $\mathscr{T} := \mathbb{R}$, and where the dot denotes derivation with respect to time t. Then $\dot{\eta}(t) := \dot{u}(t) - \dot{y}(t)$ belongs to the outward normal cone for Z at the point $y(t)$. If Z has non-empty interior, the decomposition

$$u = y + \eta$$

where η has bounded total variation, can be extended to any continuous function u. The mappings

$$u \mapsto y \qquad \text{and} \qquad u \mapsto \eta$$

are then the so-called stop and play operator, respectively, special kinds of rate-independent hysteresis operators (see Krejčí [1999]).

Evolution variational inequalities have been extracted as a common feature of different physical models; they play a central rôle in modelling nonequilibrium processes with rate-independent memory in mechanics of elastoplastic and thermoelastoplastic materials including metals, polymers, as well as ferromagnetism, piezoelectricity or phase transitions. The evolution variational inequalities are there typically interpreted as a special form of the maximal dissipation principle in evolution systems with convex constraints (compare Krejčí [1999] for further references).

Hysteresis phenomena can also be encountered in physics in superconductivity and shape memory alloys, in engineering in thermostats, porous media filtration, granular motion, semiconductors, spin glasses, mechanical damage and fatigue; hysteresis also appears in chemistry, biology, economics, and even in experimental psychology (Visintin [1994]).

History The term hysteresis seems to be used for the first time by J.A. Ewing (1882) in his studies of ferromagnetism (we follow Visintin [1994]), but the phenomenon was known already to the pioneers of ferromagnetism, Weber, Maxwell, Wiedemann. In 1887, Lord Rayleigh proposed a model of ferromagnetic hysteresis which is now called Prandtl-Ishlinskiĭ model of play-type. The so-called Preisach model was actually proposed by Weiss and Freudenreich (1916), Preisach revisited this idea in 1935 and introduced the geometrical interpretation. This construction is one of the main features of the model. Already in 1924, Prandtl introduced a scalar model of elasto-plasticity which is now known as linear stop and was later extended to tensors by Reuss (Prandtl-Reuss model). In 1928, Prandtl proposed a much more general model, obtained by composing a family of linear stops, the already mentioned Prandtl-Ishlinskiĭ model. The mathematical history of hysteresis is much shorter: it seems that the first mathematical treatment was in 1966 by Bouc, an engineering student, who regarded hysteresis as a map between function spaces. In 1970, Krasnosels'skiĭ and co-workers proposed a mathematical formulation of the Prandtl-Ishlinskiĭ model, in terms of hysteresis operators. Then, in the years 1970-1980, Krasnosel'skiĭ, Prokovskiĭ and others conducted a systematic study of the mathematical properties of these operators, finally leading to the monograph Krasnosel'skiĭ and Pokrovskiĭ [1989], first published in Russian in 1983.

Input/output diagram We want to consider only rate independent hysteresis. Rate independence means, as described under the name "range invariance" for dynamical systems in the previous section 2.1, that for every time transformation ψ, inputs u resp. $\tilde{u}$ and corresponding outputs y resp. $\tilde{y}$ of the hysteresis system, the equality $u = \tilde{u} \circ \psi$ implies the equality $y = \tilde{y} \circ \psi$. This especially means that the velocity of the signals is not recognized by the system. Therefore, the rate independence allows for a graphical representation as u-y-diagram, an example is shown in figure 2.3.

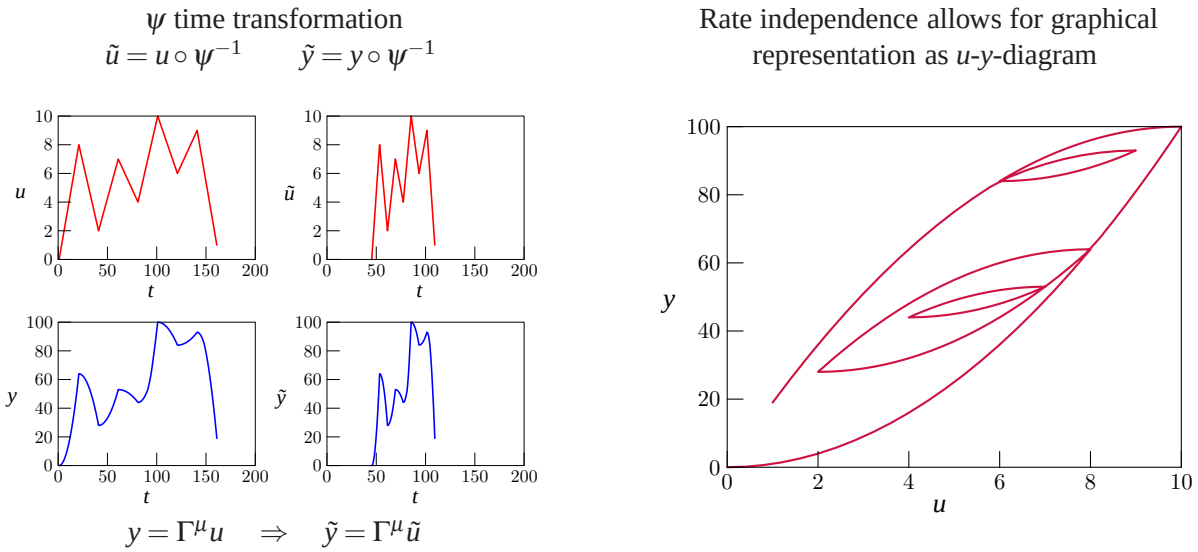


Figure 2.3: Original and transformed inputs and outputs (left) result in the same input/output diagram (right)

Piecewise monotone functions Before we define special hysteresis operators in a formal way, we have to define the space where the input functions $u(\cdot)$ live in (see e.g. Brokate and Sprekels [1996]):

Definition 2.8: **(1)** *Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be a function. We call the $(n+1)$-tuple*

$$(t_0, \ldots, t_n) \qquad \text{with } n \in \mathbb{N},\ t_0 < \cdots < t_n \in \mathbb{R},$$

*a **monotonicity partition** of f, if for all $k = 0, \ldots, n$ the function f is monotone on all intervals*

$$I_{-1} := (-\infty, t_0], \quad I_0 := [t_0, t_1], \quad \ldots, \quad I_{n-1} := [t_{n-1}, t_n], \quad I_n := [t_n, \infty),$$

i.e. for all intervals I_k, $k = -1, \ldots, n$, holds

$$f(t) \leq f(\tau) \quad \text{or} \quad f(\tau) \leq f(t) \quad \text{for all } t, \tau \in I_k \text{ with } t < \tau.$$

*If such a partition exists, then we call f a **piecewise monotone function**.*

(2) *Let M_{pm} denote the vector space of all **piecewise monotone functions** $f : \mathbb{R} \longrightarrow \mathbb{R}$:*

$$M_{\mathrm{pm}} := \left\{ f : \mathbb{R} \longrightarrow \mathbb{R} \;\middle|\; \text{there exists a monotonicity partition of } f \right\}.$$

(3) *Let C_{pm} denote the vector space of all **continuous piecewise monotone functions** $f : \mathbb{R} \longrightarrow \mathbb{R}$:*

$$C_{\mathrm{pm}} := M_{\mathrm{pm}} \cap C^0(\mathbb{R}).$$

($C^0(\mathbb{R})$ denotes the Banach space of continuous and bounded real or complex functions on $\mathbb{R}$, see appendix.)

Remark: For functions $f \in M_{\text{pm}}$ there exists always a monotonicity partition which is minimal. If we build equivalence classes of all functions with the same minimal monotonicity partition, each equivalence class will contain at least one element of C_{pm}. Thus, the equivalence relation on M_{pm} and on C_{pm} leads to the same quotient spaces.

The usual way to define hysteresis operators is to define them first on C_{pm} and afterwards to extend the domain to some (subspace of) $C^0(\mathbb{R})$. Nevertheless, the Preisach model can be defined on $C^0(\mathbb{R})$ directly.

Delayed relays The Preisach model, that we want to present in the next paragraphs, can be seen as a generalized local model network. First, we need a family of basis models, also called atoms; especially in the case of a hysteresis model, an atom is called ***hysteron***. These basis models are the building blocks for the more complex hysteresis models. The hysterons for the Preisach model are the delayed relays. The use of other hysterons leads to other types of hysteresis models (e.g. the Prandtl model by taking the play operator). We define the delayed relay as a deterministic state space system:

Definition 2.9: *Let $\alpha, \beta \in \mathbb{R}$ with $\alpha < \beta$. A deterministic state space system $\Sigma = \Sigma^{\alpha,\beta} = (\mathscr{T}, \mathscr{U}, \mathscr{U}^*, \mathscr{X}, \mathscr{Y}, \varphi, \eta)$ is called **delayed relay** with **lower threshold** α and **upper threshold** β, if the following holds:*

- $\mathscr{T} := \mathbb{R}$ *or* $\mathscr{T} := \mathbb{Z}$,
- $\mathscr{U} := \mathbb{R}$,
- $\mathscr{U}^* = C^0(\mathbb{R})$,
- $\mathscr{X} := \mathscr{Y} := \{-1, +1\}$
- $$\varphi(t; t_0, x_0, u(\cdot)) := \begin{cases} +1 & \text{if there exists } t_1 \in [t_0, t) \text{ such that } u(t_1) \geq \beta \\ & \text{and for all } \tau \in (t_1, t)\text{: } u(\tau) > \alpha \\ -1 & \text{if there exists } t_1 \in [t_0, t) \text{ such that } u(t_1) \leq \alpha \\ & \text{and for all } \tau \in (t_1, t)\text{: } u(\tau) < \beta \\ x_0 & \text{else} \end{cases}$$
- $\eta(t, x, u) := x$.

The state x and thus the output y will take values in the set $\{-1, +1\}$. At a time $t > t_0$ the value $y(t)$ depends on the past values of the relay and on the input values $u : [0, t) \to \mathbb{R}$: The relay will change its value from -1 to $+1$ when the input value $u(t)$ is increasing and exceeds the threshold β. On the other hand, the relay will change its value from $+1$ to -1 when the input value is decreasing and goes below the threshold α. In all other cases, the relay does not change its state and output (see figure 2.4).

We can immediately verify the following properties:

Lemma 2.2: *Let $\alpha < \beta \in \mathbb{R}$ and $\Sigma = \Sigma^{\alpha,\beta}$ be a delayed relay with thresholds α and β. Then:*

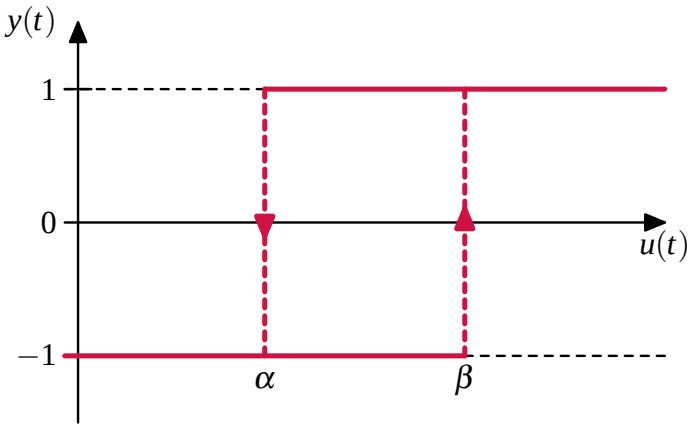


Figure 2.4: Relay operator with hysteresis

- Σ *is a complete dynamical system:* $\mathscr{T}_{t_0,x_0,u(\cdot)} = \mathscr{T}_{t_0}$,
- Σ *is nonlinear,*
- Σ *is rate independent,*
- Σ *has local memory.* □

Since the delayed relay is a complete system, the corresponding input-output operator exists for all thresholds $\alpha < \beta$, every initial time t_0, and every initial value $x_0 \in \{-1,+1\}$:

$$y(t) := (\Gamma^{\alpha,\beta}_{t_0,x_0} u(\cdot))(t) := \eta(t,x(t),u(t)) = x(t).$$

Preisach half plane

Definition 2.10: *The* ***Preisach half plane*** $\mathscr{P}$ *is given by the set of all admissible thresholds* (α,β)*:*

$$\mathscr{P} := \{(\alpha,\beta) \in \mathbb{R}^2 \mid \alpha < \beta\}$$

(see figure 2.5).

The Preisach model is then given as a weighted parallel superposition of all relays with admissible thresholds, where the weighting is done by a finite signed Borel measure μ on the Preisach half plane $\mathscr{P}$. Recall that the Borel σ-algebra $\mathfrak{B}$ on a topological space Ω is generated by the compact subsets of Ω. An (unsigned) measure $\mu : \mathfrak{B} \longrightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$ is then a finite (unsigned) Borel measure if it is a measure on $\mathfrak{B}$ and

$$\mu(\Omega) < \infty$$

holds. A signed measure $\mu : \mathfrak{B} \longrightarrow \mathbb{R} \cup \{-\infty,+\infty\}$ obeys the same axioms as an unsigned measure, without the restriction that the values are not allowed to be negative. But instead, for

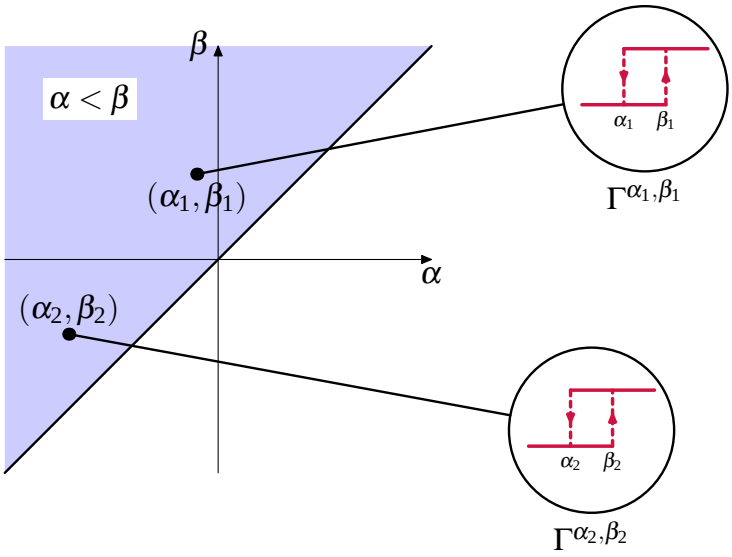


Figure 2.5: The Preisach half plane

a signed measure, one demands that only one of the values $+\infty$ or $-\infty$ is taken by μ. A signed measure μ can always be decomposed into two unsigned measures μ_+ and μ_- such that

$$\mu = \mu_+ - \mu_-.$$

In our case, we want the signed measure μ to be finite, i.e. neither of the values $-\infty$ and $+\infty$ is taken, which is equivalent to

$$\mu_+(\mathscr{P}) < \infty \qquad \text{and} \qquad \mu_-(\mathscr{P}) < \infty.$$

Preisach model

Definition 2.11: *Let μ be a finite signed Borel measure on the Preisach half plane $\mathscr{P}$. Let for any $t_0 \in \mathbb{R}$ and any μ-measurable map $x_0 : \mathscr{P} \to \{-1,+1\}$, $(\alpha,\beta) \mapsto x_0(\alpha,\beta)$, be*

$$\varphi_{\alpha,\beta}(t;t_0,x_0(\alpha,\beta),u(\cdot)) := \begin{cases} +1 & \text{if there exists } t_1 \in [t_0,t) \text{ such that } u(t_1) \geq \beta \\ & \text{and for all } \tau \in (t_1,t)\text{: } u(\tau) > \alpha \\ -1 & \text{if there exists } t_1 \in [t_0,t) \text{ such that } u(t_1) \leq \alpha \\ & \text{and for all } \tau \in (t_1,t)\text{: } u(\tau) < \beta \\ x_0(\alpha,\beta) & \text{else} \end{cases}$$

*the transition map of the delayed relay with thresholds α and β. Then the **Preisach model** with **Preisach weight** μ is given by the deterministic state space system*

$$\Sigma = \Sigma_\mu(\mathscr{T},\mathscr{U},\mathscr{U}^*,\mathscr{X},\mathscr{Y},\varphi,\eta)$$

with

- $\mathscr{T} := \mathbb{R}$ *or* $\mathscr{T} := \mathbb{Z}$,
- $\mathscr{U} := \mathbb{R}$,
- $\mathscr{U}^* = C^0(\mathbb{R})$,
- $\mathscr{X} := \{x : \mathscr{P} \longrightarrow \{-1,+1\}$ *μ-measurable*$\}$,
- $\mathscr{Y} := \mathbb{R}$,
- $\varphi(t;t_0,x_0,u(\cdot))(\alpha,\beta) := \varphi_{\alpha,\beta}(t;t_0,x_0(\alpha,\beta),u(\cdot))$,
- $\eta(t,x,u) := \int_{(\alpha,\beta)\in\mathscr{P}} x(\alpha,\beta)d\mu(\alpha,\beta)$.

If we compare this definition to the definition of a generalized local model network, we recognize that the Preisach model is nothing else than a special case of a generalized local model network where the partial models are given by delayed relays.

Properties of the Preisach model We begin with some immediate observations:

Lemma 2.3: *Let μ be a finite Borel measure on the Preisach half plane $\mathscr{P}$ and let $\Sigma = \Sigma^\mu$ be the Preisach model with Preisach weight μ. Then it holds that:*

- Σ *is a complete dynamical system:* $\mathscr{T}_{t_0,x_0,u(\cdot)} = \mathscr{T}_{t_0}$,
- Σ *is nonlinear,*
- Σ *is rate independent,*
- Σ *has* nonlocal memory.

The last property is an important new property compared to the delayed relay which only shows a local memory.

Since each Preisach model Σ^μ is complete as a deterministic state space system, we can define the ***Preisach operator*** $\Gamma^\mu_{t_0,x_0}$ on functions of $C^0(\mathbb{R})$ for each $t_0 \in \mathscr{T}$ and each μ-measurable $x_0 : \mathscr{P} \longrightarrow \{-1,+1\}$ to be the input/ouput-operator of Σ^μ:

$$y(t) := (\Gamma^\mu_{t_0,x_0}u)(t) := \eta(t,x(t),u(t)) = \int_{(\alpha,\beta)\in\mathscr{P}} x(\alpha,\beta)d\mu(\alpha,\beta).$$

It can be shown that this is actually an operator

$$\Gamma^\mu_{t_0,x_0} : C^0(\mathbb{R}) \longrightarrow L^\infty(t_0,\infty) \cap C^0_l([t_0,+\infty)),$$

where C^0_l is the space of bounded functions which are continuous on the left (see e.g. Visintin [1994].

Remark: Our choice of the Preisach operator is the alternative with delayed jumps which Visintin denotes by $\mathscr{H}^*$. There are other versions of the Preisach operator with instantaneous jumps. Our choice has been made to be in accordance with the axioms of deterministic dynamical systems by retaining a simple definition. The main point is that the states $x(t)$ only depend on earlier inputs $u(\tau)$, $\tau < t$. Thus, a direct dependence of the output $y(t)$ on the input $u(t)$ as it is given with instantaneous jumps must be modelled with the output function η. The modified Preisach operators fit as well into our framework, but the output function would require a modification.

The Preisach operator can also be seen as a continuous linear parallel superposition of relay operators, weighted by the measure μ:

$$\Gamma^{\mu}_{t_0,x_0} u = \int_{(\alpha,\beta)\in\mathscr{P}} \Gamma^{\alpha,\beta}_{t_0,x_0(\alpha,\beta)} u \, d\mu(\alpha,\beta).$$

i.e.

$$y(t) := (\Gamma^{\mu}_{t_0,x_0} u)(t) = \int_{(\alpha,\beta)\in\mathscr{P}} (\Gamma^{\alpha,\beta}_{t_0,x_0(\alpha,\beta)} u)(t) d\mu(\alpha,\beta).$$

Examples: (1) Discrete parallel superposition:

Let μ be given by a weighted finite sum (mixture) of Dirac measures: For each $B \in \mathfrak{B}$ define

$$\mu(B) := \sum_{i=1}^{n} \omega_i \delta_{(\alpha_i,\beta_i)}(B)$$

with

$$\delta_{(\alpha_i,\beta_i)}(B) := \begin{cases} 1 & \text{if } (\alpha_i,\beta_i) \in B \\ 0 & \text{else} \end{cases}$$

for given points $(\alpha_i,\beta_i) \in \mathscr{P}$ and weights $\omega_i \in \mathbb{R}$, $i = 1,\ldots,n$ and $n \in \mathbb{N}$. Then:

$$y(t) := (\Gamma^{\mu}_{t_0,x_0} u)(t) = \int_{(\alpha,\beta)\in\mathscr{P}} (\Gamma^{\alpha,\beta}_{t_0,x_0(\alpha,\beta)} u)(t) d\mu(\alpha,\beta) = \sum_{i=1}^{n} \omega_i \cdot (\Gamma^{\alpha_i,\beta_i}_{t_0,x_0(\alpha_i,\beta_i)} u)(t)$$

for all $t_0 \in \mathscr{T}$ and $x_0 : \mathscr{P} \longrightarrow \{-1,+1\}$ (but of course it is enough to provide the n values $x_0(\alpha_i,\beta_i)$ for $i = 1,\ldots,n$).

In the special case $n := 1$ and $\omega_1 := 1$ this reduces to

$$y(t) := (\Gamma^{\mu}_{t_0,x_0} u)(t) = (\Gamma^{\alpha_1,\beta_1}_{t_0,x_0(\alpha_1,\beta_1)} u)(t),$$

which is the relay operator with thresholds α_1 and β_1.

If in contrast $n > 1$, then we observe the occurrence of inner loops in the input/ouput diagram. The internal memory of this system is still local because the maximal number of nested inner loops is finite: one can store the information, in which of these inner loops the system actually is, with a finite vector. Thus, a finite vector is enough for the system to remember how it has reached the current state, i.e. which path it has taken; it "knows" where it has to close an inner loop and when to return to the next outer loop. A schematic view of the parallel superposition of three relays is shown in figure 2.6, the outer and one possible inner hysteresis loop can be seen in figure 2.7.

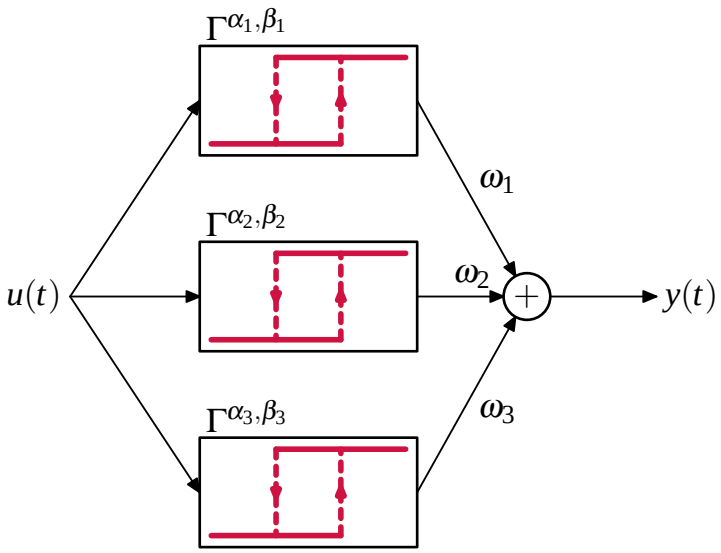


Figure 2.6: Parallel superposition of three relay operators

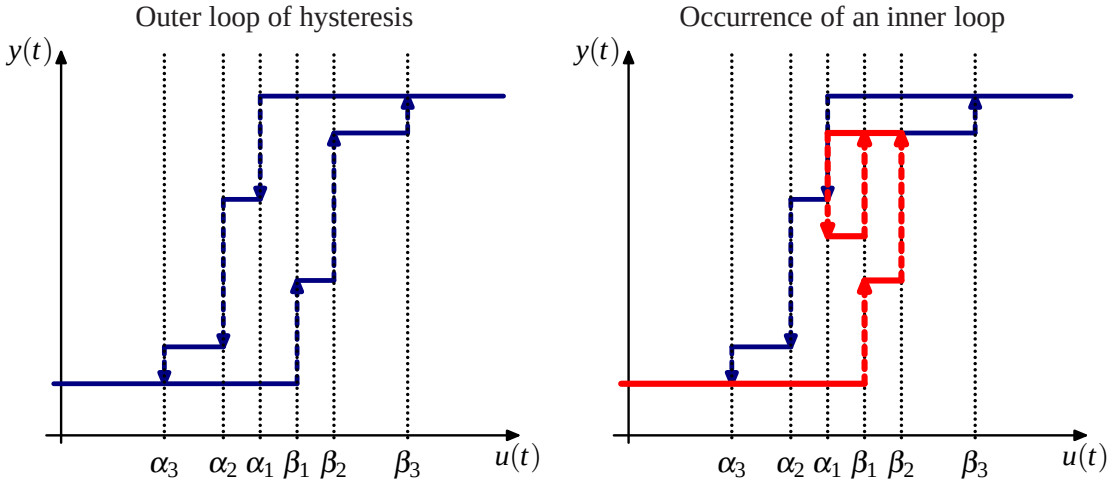


Figure 2.7: Outer and inner hysteresis loops

(2) Continuous parallel superposition:

If the measure μ is absolutely continuous with respect to the Lebesgue measure λ on $\mathscr{P}$, then μ can be related to λ by a density function

$$\omega : \mathscr{P} \longrightarrow \mathbb{R}$$

such that

$$d\mu = \omega d\lambda$$

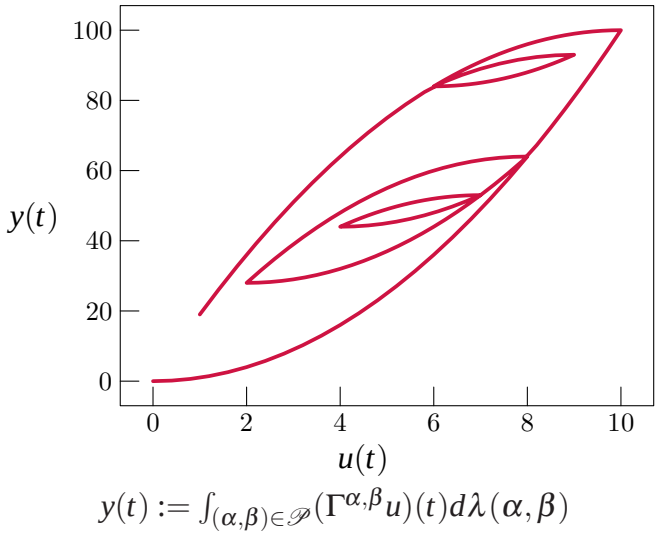
holds. In this case we obtain for the input/ouptput operator $\Gamma^\mu = \Gamma^\mu_{t_0,x_0}$:

$$\begin{aligned}\Gamma^\mu &= \int\limits_{(\alpha,\beta)\in\mathscr{P}} \Gamma^{\alpha,\beta} d\mu(\alpha,\beta) = \int\limits_{(\alpha,\beta)\in\mathscr{P}} \omega(\alpha,\beta)\Gamma^{\alpha,\beta} d\lambda(\alpha,\beta) \\ &= \iint\limits_{(\alpha,\beta)\in\mathscr{P}} \omega(\alpha,\beta)\Gamma^{\alpha,\beta} d\beta d\alpha.\end{aligned}$$

Figure 2.8 shows the input/output diagram for the case where $\omega \equiv 1$ on some large bounded area, e.g. a large triangle given by the vertices

$$(m_{-1}, m_{-1}), (m_{-1}, M_{-1}), (M_{-1}, M_{-1})$$

with $m_{-1} < M_{-1}$, and $\omega \equiv 0$ outside (remember that we required the measure μ to be finite). This example shows real nonlocal memory: The number of nested inner loops is



$y(t) := \int_{(\alpha,\beta)\in\mathscr{P}} (\Gamma^{\alpha,\beta} u)(t) d\lambda(\alpha,\beta)$

Figure 2.8: Input/output diagram of a hysteresis with density $\omega \equiv 1$ (on some large triangle)

principally infinite. The internal state of the system must remember which loop the system is in. This cannot be done with a finite-dimensional vector.

Geometric interpretation We have the following geometric interpretation of the Preisach model (see e.g. Mayergoyz [1991]): Considering the partition of the Preisach half plane at each time t into the two sets $S_+(t)$ and $S_-(t)$, defined by

$$S_\pm(t) := \big\{(\alpha,\beta) \in \mathscr{P} \;\big|\; (\Gamma^{\alpha,\beta}u)(t) = \pm 1\big\},$$

we observe that

$$\mathscr{P} = S_+(t) \dot\cup S_-(t) \qquad \text{(disjoint union)}.$$

From this, it follows for each time t that

$$\begin{aligned} y(t) &:= \int_{(\alpha,\beta)\in\mathscr{P}} (\Gamma^{\alpha,\beta}u)(t)\, d\mu(\alpha,\beta) \\ &= \int_{S_+(t)} d\mu - \int_{S_-(t)} d\mu \\ &= \int_{S_+(t)} d\mu - \Big(\int_{\mathscr{P}} d\mu - \int_{S_+(t)} d\mu\Big) \\ &= 2\int_{S_+(t)} d\mu - \int_{\mathscr{P}} d\mu. \end{aligned}$$

It is thus enough to consider solely $S_+(t)$.

For certain initial conditions there is an interesting characterization of the sets $S_+(t)$ and $S_-(t)$. Let us assume that we begin at some initial time t_0 and initial state $x_0 : \mathscr{P} \longrightarrow \{-1,+1\}$ given by $x_0(\alpha,\beta) = -1$ for all $(\alpha,\beta) \in \mathscr{P}$. This is equivalent to $S_-(t_0) = \mathscr{P}$ and $S_+(t_0) = \emptyset$: All relays $\Gamma^{\alpha,\beta}$ are initially in the state -1. Let us further assume that some input function $u \in C^0(\mathbb{R})$ first increases with increasing time $t \geq t_0$. Then the set $S_+(t)$ equals

$$\big\{(\alpha,\beta) \in \mathscr{P} \;\big|\; \beta < u(t)\big\},$$

because all relays $\Gamma^{\alpha,\beta}$ with $\beta < u(t)$ jump to the state $+1$, until at time t_1 we reach a first local maximum $M_0 := u(t_1)$ of u. After this point the value $u(t)$ decreases if times goes on, and we get

$$S_+(t) = \big\{(\alpha,\beta) \in \mathscr{P} \;\big|\; \alpha \leq u(t) \text{ and } \beta < M_0\big\}$$

until $u(t)$ reaches the first local minimum m_0. When $u(t)$ further alternatingly increases and decreases with increasing time t, the common boundary of $S_+(t)$ and $S_-(t)$ is always given by a polygon $A(t)$ with only horizontal and vertical lines as edges; furthermore, as graph, the polygon is non-increasing, see figure 2.9. The figure shows also clearly the evolution of the hysteresis loops. As we have seen above, the output $y(t)$ is equal to 2 times the measure of the shaded area (i.e. $S_+(t)$) plus a constant offset (given by the negative of the measure of the complete Preisach half plane).

The polygon $A(t)$ has exactly one infinite edge. This edge is parallel to the α-axis and infinite towards $\alpha \to -\infty$. We recall that all this is only valid if the initial state is given by $S_+(t_0) = \emptyset$. This initial state is called ***negative saturation***. A similar construction applies if the initial state is $S_+(t_0) = \mathscr{P}$. This initial state is called ***positive saturation***. In this case, the

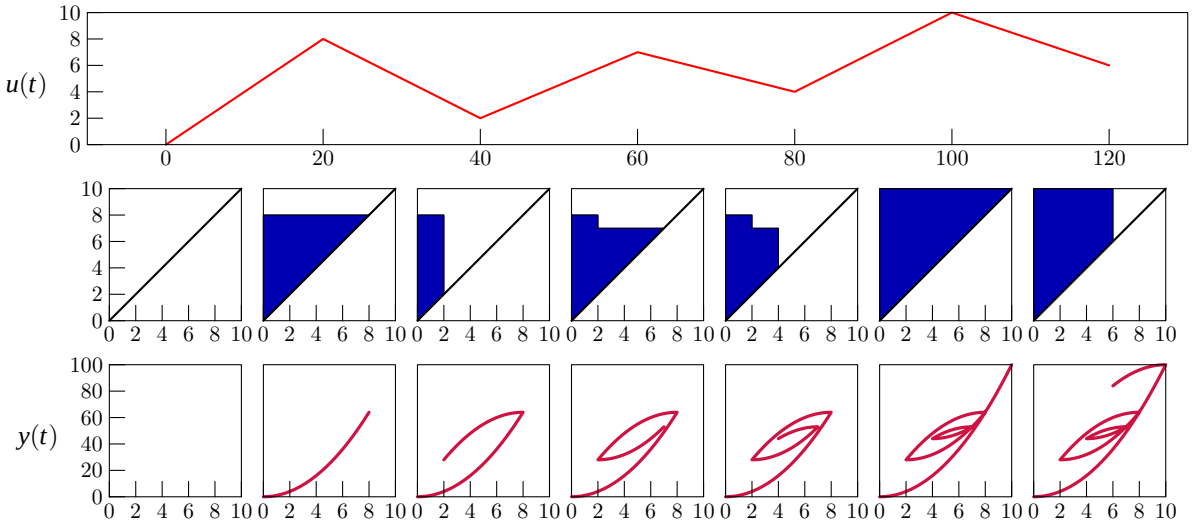


Figure 2.9: The evolution of the partitions of the Preisach half plane. First row: Model input. Second row: Partition of Preisach half plane. Third row: Input/output diagrams

common boundary $A(t)$ is also a polygon with exactly one infinite edge, but it is parallel to the β-axis and infinite towards $\beta \to +\infty$. In both cases, the number of vertices is either finite or countably infinite. If the number of the vertices is countably infinite, the vertices can only accumulate in a neighbourhood of the diagonal $\alpha = \beta$.

Apart from the mentioned two initial conditions, we could also begin with every initial condition which corresponds to a partition of the Preisach plane resulting from the above constructions for $S_+(t)$, $S_-(t)$ and $A(t)$. To fully understand the shapes of $S_+(t)$ and $S_-(t)$, we consider the memory sequences.

Reduced memory sequence as representation sequence The reduced memory sequence consists of certain dominant minima and maxima of the input u. The knowledge of this sequence is enough to reconstruct the output of the hysteresis operator. We will present the construction of such reduced memory sequences. Before that, we shortly describe the "longer" *complete* memory sequences which contain *all* extrema of the input u. We follow Visintin [1994].

The complete memory sequence cannot be defined for every function in $C^0(\mathbb{R})$. Let therefore $u \in C_{\mathrm{pm}}(\mathbb{R})$ be a continuous piecewise monotone function. For any $t \in \mathbb{R}$, consider the finite sequence $(t_j)_{j=0,\dots,m}$ of time instants at which u changes its monotonicity, such that

$$t_0 < t_1 < t_2 < \cdots < t_m = t.$$

Consider also the finite sequence of the corresponding input values $(u(t_j))_{j=1,\dots,m}$. We call this latter sequence the ***complete memory sequence*** of the function u at time instant t. It consists of alternating minima and maxima of u. The rate independence property of a hysteresis operator ensures that this sequence is enough to determine the output value of the operator at time t. Such sequences do not exist for all continuous functions, even not for all infinitely differentiable functions, and even if we allowed infinite sequences.

For the Preisach model it is enough to consider the ***reduced memory sequences***, existing (as infinite sequences) for all continuous functions $u \in C^0(\mathscr{T}_{t_0})$, the continuous and bounded functions on the time span

$$\mathscr{T}_{t_0} = \big\{ t \in \mathscr{T} \;\big|\; t \geq t_0 \big\}.$$

Let $t \in \mathscr{T}_{t_0}$ be fixed. Then we define the reduced memory sequence

$$(r_j)_{j=1,\ldots,m} := (u(s_j))_{j=1,\ldots,m}$$

for u and t corresponding to a sequence of times

$$t_0 \leq s_1 < s_2 < \cdots < s_m = t$$

by the following algorithm (with initialization slightly different from Visintin [1994]):

- Set
$$M_0 := \max_{\tau\in[t_0,t]} \big\{u(\tau)\big\}, \qquad s_1^{\max} := \max\big\{ \tau \in [t_0,t] \;\big|\; u(\tau) = M_0 \big\}$$
and
$$m_0 := \min_{\tau\in[t_0,t]} \big\{u(\tau)\big\}, \qquad s_1^{\min} := \max\big\{ \tau \in [t_0,t] \;\big|\; u(\tau) = m_0 \big\}.$$
We have $M_0 = m_0$ exactly if $s_1^{\max} = s_1^{\min}$. In this case, $M_0 = m_0 = u(t_0) = u(t)$ and $s_1^{\max} = s_1^{\min} = t$, and setting
$$s_1 := t \qquad \text{and} \qquad r_1 := u(t),$$
we are done. Else, $m_0 < M_0$, and either $s_1^{\min} < s_1^{\max}$ or $s_1^{\max} < s_1^{\min}$. In the first case, we set
$$s_1 := s_1^{\min}, \qquad r_1 := m_0, \qquad \text{and} \qquad s_2 := s_1^{\max}, \qquad r_2 := M_0.$$
In the second case, we set
$$s_1 := s_1^{\max}, \qquad r_1 := M_0, \qquad \text{and} \qquad s_2 := s_1^{\min}, \qquad r_2 := m_0.$$
If $s_2 = t$ we are done. Else, in both cases, $t_0 \leq s_1 < s_2$ and $m_0 < u(t) < M_0$ holds, and we have either $u(s_2) = M_0$ or $u(s_2) = m_0$.

- Assume now inductively, for any $k \in \mathbb{N}$, that $t_0 \leq s_1 < \ldots < s_{2(k+1)}$ and
$$m_0 < m_1 < \cdots < m_k < u(t) < M_k < \cdots < M_1 < M_0$$
are already given, and we have either $u(s_{2(k+1)}) = M_k$ or $u(s_{2(k+1)}) = m_k$.

- If $u(s_{2(k+1)}) = M_k$, then set
$$m_{k+1} := \min_{\tau\in[s_{2(k+1)},t]} \big\{u(\tau)\big\}$$

and

$$r_{2(k+1)+1} := m_{k+1}, \qquad s_{2(k+1)+1} := \max\big\{\tau \in [s_{2(k+1)}, t] \;\big|\; u(\tau) = m_{k+1}\big\},$$

else set

$$M_{k+1} := \max_{\tau\in[s_{2(k+1)},t]} \big\{u(\tau)\big\}$$

and

$$r_{2(k+1)+1} := M_{k+1}, \qquad s_{2(k+1)+1} := \max\big\{\tau \in [s_{2(k+1)}, t] \;\big|\; u(\tau) = M_{k+1}\big\}.$$

If $s_{2(k+1)+1} = t$, we are done. Else:

- If $u(s_{2(k+1)}) = M_k$, then set

$$M_{k+1} := \max_{\tau\in[s_{2(k+1)+1},t]} \big\{u(\tau)\big\}$$

and

$$r_{2(k+1)+2} := M_{k+1}, \qquad s_{2(k+1)+2} := \max\big\{\tau \in [s_{2(k+1)+1}, t] \;\big|\; u(\tau) = M_{k+1}\big\},$$

else set

$$m_{k+1} := \min_{\tau\in[s_{2(k+1)+1},t]} \big\{u(\tau)\big\}$$

and

$$r_{2(k+1)+2} := m_{k+1}, \qquad s_{2(k+1)+2} := \max\big\{\tau \in [s_{2(k+1)+1}, t] \;\big|\; u(\tau) = m_{k+1}\big\}.$$

If $s_{2(k+1)+2} = t$, we are done. Else, set $k \leftarrow k+1$ and repeat the last two steps.

If this algorithm does not stop, i.e. the sequence (s_j) is infinite, then we see that

$$t_0 \le s_1 < s_2 < \cdots < s_j < \cdots < t$$

and the reduced memory sequence (r_j) for u and t is then given by either

$$(r_j)_{j\ge1} = (M_0, m_0, M_1, m_1, M_2, m_2, \ldots)$$

or

$$(r_j)_{j\ge1} = (m_0, M_0, m_1, M_1, m_2, M_2, \ldots)$$

with

$$m_0 < m_1 < m_2 < \cdots < m_k < \cdots < u(t) < \cdots < M_k < \cdots < M_2 < M_1 < M_0.$$

Setting

$$s^* = \sup\big\{s_j\big\}$$

we have that $s^* \leq t$ and u is constant in $[s^*, t]$, and

$$\lim_{k\to\infty} m_k = \lim_{k\to\infty} M_k = u(s^*) = u(t).$$

Thus, (m_k) is a strictly increasing sequence of local minima, and (M_k) is a strictly decreasing sequence of local maxima, and the sequence

$$\big(|r_{j+1} - r_j|\big)_{j\geq 1} = \big(|u(s_{j+1}) - u(s_j)|\big)_{j\geq 1}$$

is strictly decreasing. The case of a finite number of steps is similar. The reduced memory sequence is then finite and given by either

$$(r_j)_{j=1,\ldots,m} = (M_0, m_0, M_1, m_1, M_2, m_2, \ldots, u(t))$$

or

$$(r_j)_{j=1,\ldots,m} = (m_0, M_0, m_1, M_1, m_2, M_2, \ldots, u(t)),$$

where in both cases we have the possibilities that $r_m = u(t)$ is equal to the last local minimum or to the last local maximum. We remark that the reduced memory sequence is finite if $u \in C_{\mathrm{pm}}(\mathscr{T}_{t_0})$, the continuous piecewise monotone functions on $\mathscr{T}_{t_0}$. The converse does not hold. Examples of reduced memory sequences for several times t and an input $u \in C_{\mathrm{pm}}(\mathscr{T}_{t_0})$ with $t_0 = 0$ is given in figure 2.10.

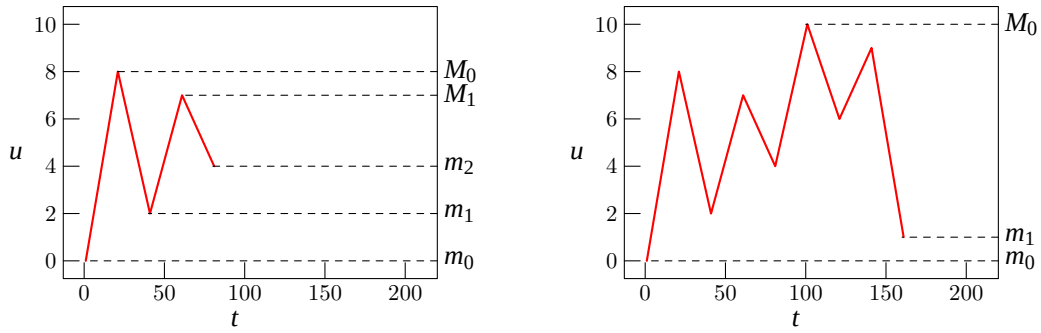


Figure 2.10: Reduced memory sequences at time $t = 80$ (left) and $t = 160$ (right)

Alternative characterization of reduced memory sequences The image of the map defined by the previous algorithm for all $u \in C^0(\mathscr{T}_{t_0})$ on the set of sequences of $\mathbb{R}$ is the set of alternating sequences $\mathscr{S} := \mathscr{S}(\mathbb{R})$ over $\mathbb{R}$. We can also define this set in the following way: $\mathscr{S} := \mathscr{S}(\mathbb{R})$ is the set of ***alternating sequences*** over $\mathbb{R}$ if

$$\mathscr{S} := \bigcup_{m\in\mathbb{N}\cup\{\infty\}} \mathscr{S}_m$$

and where each $\mathscr{S}_m$, $m \in \mathbb{N}$ is recursively defined by

$$\mathscr{S}_0 := \{\emptyset\}, \qquad \mathscr{S}_1 := \mathbb{R}, \qquad \mathscr{S}_2 := \mathbb{R}^2 \setminus \mathrm{diag}\{\mathbb{R}^2\},$$

and

$$\mathscr{S}_{m+1} := \big\{ r = (r_1, \dots, r_{m+1}) \,\big|\, (r_1, \dots, r_m) \in \mathscr{S}_m,\, r_{m+1} \in \mathbb{R}, \\ \text{and either } r_{m-1} < r_{m+1} < r_m \text{ or } r_{m-1} > r_{m+1} > r_m \big\}$$

for $2 < m \in \mathbb{N}$, and $\mathscr{S}_\infty$ is defined as the projective limit (on the category of sets and maps) over the $\mathscr{S}_m$, $m \in \mathbb{N}$,

$$\mathscr{S}_\infty := \varprojlim \mathscr{S}_m,$$

i.e. as the set of sequences $(r_j)_{j\in\mathbb{N}}$ such that each finite "head" $(r_j)_{j=1,\dots,m}$ belongs to $\mathscr{S}_m$.

As it follows directly from this definition, each alternating sequence $r = (r_1, \dots, r_m)$ of length m begins with (M_0, m_0) or (m_0, M_0) where $m_0 < M_0$, and per induction it follows $m_0 < r_j < M_0$ for all $j > 2$. This means that m_0 is the absolute minimum and M_0 is the absolute maximum of all the values r_j, $j \in \mathbb{N}$. The same reasoning applies to the tails $(r_j, r_{j+1}, \dots)$, $j \in \mathbb{N}$. Therefore we are able to write

$$r = (r_1, r_2, \dots) =: (M_0, m_0, M_1, m_1, M_2, m_2, \dots) \quad \text{or} \quad (m_0, M_0, m_1, M_1, m_2, M_2, \dots)$$

with

$$m_0 < m_1 < m_2 < \cdots < M_2 < M_1 < M_0.$$

This is the reason for calling these sequences alternating sequences. The alternating sequences without $\mathscr{S}_0$ are in one-to-one correspondence with the reduced memory sequences of functions $u \in C^0(\mathscr{T}_{t_0})$. The previous algorithm provides a surjective map from $C^0(\mathscr{T}_{t_0})$ onto $\mathscr{S}(\mathbb{R}) \setminus \mathscr{S}_0(\mathbb{R})$ for all $t_0 \in \mathbb{R}$: All sequences (r_j) produced by this algorithm are alternating, and, for a given alternating sequence (r_j), it is easy to construct a function $u \in C^0(\mathscr{T}_{t_0})$ by $u(t_0 + j) := r_j$ and monotone interpolation which yields (r_j) as assigned reduced alternating sequence.

Prefixed reduced memory sequence If we take a look at the Preisach plane $\mathscr{P}$ and its partition into $S_+(t)$ and $S_-(t)$ with their common boundary $A(t)$, we see that the edges of this polygon agree exactly with the coefficients of the corresponding reduced memory sequence. Furthermore, we are aware that the polygon $A(t)$ is limited to the right and downwards by the diagonal $\alpha = \beta$. To the left and upwards, the polygon $A(t)$ is not bounded. Since the vertices can only accumulate at neighbourhoods of the diagonal $\alpha = \beta$, there are two possibilities:

- Either, there is a halfline which is parallel to the α-axis and infinite to the left ($\alpha \to -\infty$); then the first edge of the polygon denotes a minimum.
- Or, there is a halfline which is parallel to the β-axis and infinite upwards ($\beta \to \infty$); then the first edge of the polygon denotes a maximum.

If we construct the polygon $A(t_1)$ for any later time $t_1 > t$, we see that the decision whether the first edge is a minimum or a maximum cannot be changed anymore. The corresponding alternating sequence usually reflects this: if $r_1 < r_2$ then r_1 is a minimum, if $r_1 > r_2$ then r_1 is a maximum. A problem occurs when $A(t)$ is a polygon with less than two edges. Then the corresponding alternating sequence has less than length 2 and it is ambiguous if it should begin with a maximum or a minimum.

Therefore, we construct the prefixed alternating sequences $\overline{\mathscr{S}} := \overline{\mathscr{S}}(\overline{\mathbb{R}})$ over the affinely extended real numbers $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$:

$$\overline{\mathscr{S}} := \bigcup_{m \in \mathbb{N} \cup \{\infty\}} \overline{\mathscr{S}}_m$$

where each $\overline{\mathscr{S}}_m$ is recursively defined by

$$\overline{\mathscr{S}}_0 := \{(-\infty, +\infty), (+\infty, -\infty)\}$$

and

$$\overline{\mathscr{S}}_{m+1} := \{r = (r_{-1}, r_0, r_1 \ldots, r_{m+1}) \mid (r_{-1}, r_0, r_1, \ldots, r_m) \in \overline{\mathscr{S}}_m,\ r_{m+1} \in \mathbb{R},\ \text{and either } r_{m-1} < r_{m+1} < r_m \text{ or } r_{m-1} > r_{m+1} > r_m\}$$

for $0 < m \in \mathbb{N}$, and $\overline{\mathscr{S}}_\infty$ is again the projective limit of the $\overline{\mathscr{S}}_m$, $m \in \mathbb{N}$. We see from the definition that all alternating sequences (r_j) have at least length 2 and that they begin with either

$$(r_{-1}, r_0) = (-\infty, +\infty) \qquad \text{or} \qquad (r_{-1}, r_0) = (+\infty, -\infty),$$

whereas the tails $(r_1, r_2, \ldots)$ are exactly the alternating sequences from $\mathscr{S}(\mathbb{R})$.

Description of $S_+(t)$ by prefixed reduced memory sequences We come back to the disjoint partition of the Preisach plane $\mathscr{P}$ by $S_+(t)$ and $S_-(t)$ at each time $t \geq t_0$. Given a prefixed reduced memory sequence $(r_j)_{j \geq -1}$, we denote by r_m either the last element in (r_j) if this sequence is finite of length $m+2$ or $r_m := r_\infty := \lim_{k \to \infty} m_k = \lim_{k \to \infty} M_k$ if (r_j) is infinite. We define the polyhedron $S_+((r_j))$ to be the unbounded polyhedron with vertices

$$(-\infty, -\infty),\ (-\infty, +\infty),\ (m_0, +\infty),\ (m_0, M_0),\ (m_1, M_0),\ (m_1, M_1), \ldots,\ (r_m, r_m)$$

i.e.

$$(r_{-1}, r_{-1}),\ (r_{-1}, r_0),\ (r_1, r_0),\ (r_1, r_2),\ (r_3, r_2),\ (r_3, r_4), \ldots,\ (r_m, r_m)$$

if $(r_j) = (-\infty, +\infty, m_0, M_0, m_1, M_1, \ldots)$, and with vertices

$$(-\infty, -\infty),\ (-\infty, M_0),\ (m_0, M_0),\ (m_0, M_1),\ (m_1, M_1),\ (m_1, M_2), \ldots,\ (r_m, r_m)$$

i.e.

$$(r_0, r_0),\ (r_0, r_1),\ (r_2, r_1),\ (r_2, r_3),\ (r_4, r_3),\ (r_4, r_5), \ldots,\ (r_m, r_m)$$

if $(r_j) = (+\infty, -\infty, M_0, m_0, M_1, m_1, \ldots)$. Precisely, the polyhedron shall include the open inner area defined by these vertices, exclude all vertices and edges themselves except the vertical edges at the right of the polyhedron with their corresponding lower vertex (if it belongs to $\mathscr{P}$), i.e. $S_+((r_j))$ is the open polyhedron given by the vertices from above plus the sets

$$\{(\alpha, \beta) \in \mathscr{P} \mid \alpha = r_{j-1} \text{ and } r_j \leq \beta < r_{j-2}\}$$

for all $j \geq 0$ with

$$r_{j-2} > r_j > r_{j-1}.$$

A careful look at the definition of the Preisach operator shows that for each input function $u \in C^0(\mathscr{T}_{t_0})$ and corresponding prefixed reduced memory sequence (r_j) for some time $t \in \mathscr{T}_{t_0}$, we have that

$$S_+(t) = S_+((r_j)),$$

provided the initial condition fulfills $S_+(t_0) = S_+((r_{-1}, r_0))$.

Description of $S_-(t)$ by prefixed reduced memory sequences The definition of the polyhedron $S_-((r_j))$ is in some sense dual to the one for $S_+((r_j))$. The vertex $(-\infty, -\infty)$ has to be replaced by $(+\infty, +\infty)$, and correspondingly the vertex $(-\infty, +\infty)$ removed or inserted. Additionally, the polyhedron $S_-((r_j))$ includes/excludes the edges and vertices complementary to the ones of $S_+((r_j))$, such that $S_-(t) = S_-((r_j))$ and we have the disjoint union

$$S_+((r_j)) \dot{\cup} S_-((r_j)) = \mathscr{P}.$$

Representation theorem We can now intuitively understand the following properties of Preisach operators (see Mayergoyz [1991]):

Theorem 2.1: *Let μ be a Preisach weight and $\Gamma^\mu = \Gamma^\mu_{t_0,x_0}$ a Preisach operator. Then Γ^μ has the following three properties:*

- ***Rate independence**: The "velocity" of the input signal does not change the behaviour of the system: we are able to plot the input-output diagram.*
- ***Wiping-out property**: Only* dominant *local maxima and minima of the input signal count: the reduced memory sequences store the complete information contained in the system.*
- ***Congruency property**: Minor hysteresis loops occurring between the same consecutive extremal values have congruent shapes: this property ensures the correctness of the summation formulas involving primitive functions (see next section).*

The congruency property still needs some explanation: Let $u_1(t)$ and $u_2(t)$ be two inputs having different histories and thus different reduced memory sequences $(r_j^{(1)})$, $(r_j^{(2)})$, respectively. However, if starting at some time t_1 the inputs vary up and down between the same two consecutive extremal values u_+ and u_-, then the reduced memory sequences end in both cases equally with these two values:

$$(r_j^{(1)}) = (r_1^{(1)}, r_2^{(1)}, \ldots, u_+, u_-), \qquad (r_j^{(2)}) = (r_1^{(2)}, r_2^{(2)}, \ldots, u_+, u_-),$$

or

$$(r_j^{(1)}) = (r_1^{(1)}, r_2^{(1)}, \ldots, u_-, u_+), \qquad (r_j^{(2)}) = (r_1^{(2)}, r_2^{(2)}, \ldots, u_-, u_+).$$

Thus this variation results in minor hysteresis loops which are congruent. This is clear from looking at the Preisach plane, where in both cases the same triangle is appended to or subtracted from $S_+(t)$. In the next section, the summation formulas involving the primitive functions concerning these triangles will show that in both cases these series end with the same summand.

Reversely, these three properties characterize the Preisach operators completely. This is the content of the ***Mayergoyz representation theorem*** ([Mayergoyz, 1991]):

Theorem 2.2 (Mayergoyz representation theorem)**:** *The three properties*

- *rate independence,*
- *wiping-out property, and*
- *congruency of minor loops*

constitute necessary and sufficient conditions for a deterministic dynamical system on the set of input functions from $C^0(\mathscr{T}_{t_0})$, $t_0 \in \mathbb{R}$, to be represented by the Preisach model.

2.2.2 Implementation

We know:

$$(\Gamma^\mu u)(t) = 2 \int\limits_{S_+(t)} d\mu - \int\limits_{\mathscr{P}} d\mu = F(S_+(t)) - F(\mathscr{P}),$$

with

$$F(S_+(t)) := \int\limits_{S_+(t)} d\mu \qquad \text{and} \qquad F(\mathscr{P}) := \int\limits_{\mathscr{P}} d\mu \quad \text{(constant!)}.$$

How can we avoid computing integrals? The answer is: Use (some kind of) primitive function of the Preisach measure μ. We want to develop such primitive functions in the following.

Primitive shapes The special shape of $S_+(t)$ gives us the ability to divide $S_+(t)$ into simpler-shaped areas. We consider three possible partitions into such primitive shapes. The primitive shapes are (compare figure 2.11):

- Trapezoids $Q(\alpha_1, \beta, \alpha_2)$,
- Corners $C(\alpha, \beta)$, and
- Triangles $T(\alpha, \beta)$.

Following this order, we eventually will use especially the partitions into triangles, leading to the easiest representation of the Preisach measure. Indeed, associated to each of the primitive shapes S we have a ***primitive function***

$$F(S) := \int_S d\mu$$

which can be used to compute the Preisach operator. We begin with the definition of the primitive shapes.

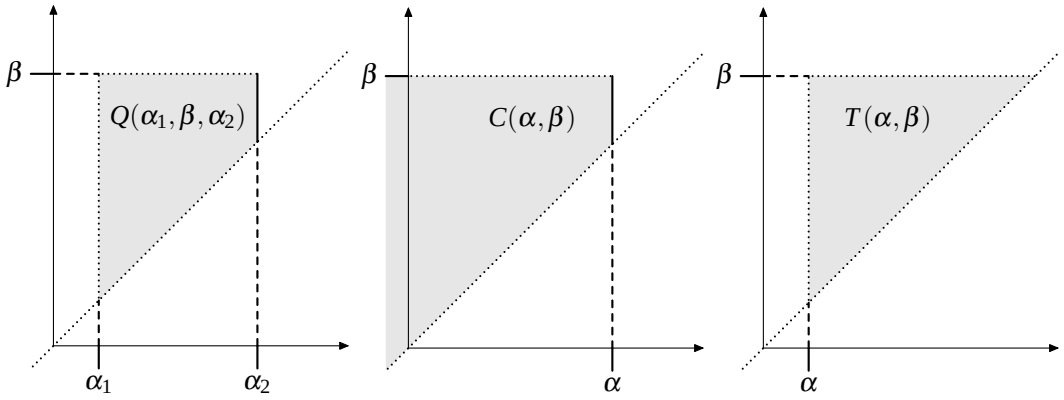


Figure 2.11: Primitive shapes

Definition 2.12 (Primitive shapes)**:** *Let $\mathscr{P}$ be the Preisach half plane. Then we define:*

- *For each $-\infty \le \alpha_1 \le \alpha_2 \le \beta \le +\infty$, the **trapezoid** $Q(\alpha_1,\beta,\alpha_2)$ with vertices*

$$(\alpha_1,\alpha_1),\ (\alpha_2,\alpha_2),\ (\alpha_2,\beta),\ (\alpha_1,\beta)$$

  is

$$Q(\alpha_1,\beta,\alpha_2) := \big\{(\tilde\alpha,\tilde\beta) \in \mathscr{P} \;\big|\; \alpha_1 < \tilde\alpha \le \alpha_2 \text{ and } \tilde\alpha < \tilde\beta < \beta\big\}.$$

- *For each $-\infty \le \alpha \le \beta \le +\infty$, the **corner** $C(\alpha,\beta)$ is*

$$C(\alpha,\beta) := \big\{(\tilde\alpha,\tilde\beta) \in \mathscr{P} \;\big|\; \tilde\alpha \le \alpha \text{ and } \tilde\alpha < \tilde\beta < \beta\big\}.$$

- *For each $-\infty \le \alpha \le \beta \le +\infty$, the **triangle** $T(\alpha,\beta)$ with vertices*

$$(\alpha,\alpha),\ (\beta,\beta),\ (\alpha,\beta)$$

  is

$$T(\alpha,\beta) := \big\{(\tilde\alpha,\tilde\beta) \in \mathscr{P} \;\big|\; \alpha < \tilde\alpha < \tilde\beta < \beta\big\}.$$

*The corresponding **primitive functions** are given by*

$$F_Q(\alpha_1,\beta,\alpha_2) := F_Q^\mu(\alpha_1,\beta,\alpha_2) := \int_{Q(\alpha_1,\beta,\alpha_2)} d\mu,$$
$$F_C(\alpha,\beta) := F_C^\mu(\alpha,\beta) := \int_{C(\alpha,\beta)} d\mu,$$
$$F_T(\alpha,\beta) := F_T^\mu(\alpha,\beta) := \int_{T(\alpha,\beta)} d\mu.$$

All vertices and the left and upper edges of all primitive shapes are excluded, whereas the right edges (without vertices) belong to the shapes. This guarantees disjoint unions of several adjoining shapes. The shapes are designed to fit to $S_+(t)$. A dual definition fitting to $S_-(t)$ would require including the upper edges instead of the right ones.

Relations between primitive functions The primitive functions can be defined by each other, as the following theorem says:

Theorem 2.3: *Let F_Q, F_C, and F_T be the primitive functions as defined above. Then:*

(i) $F_Q(\alpha_1,\beta,\alpha_2) = F_C(\alpha_2,\beta) - F_C(\alpha_1,\beta) = F_T(\alpha_1,\beta) - F_T(\alpha_2,\beta)$,

(ii) $F_C(\alpha,\beta) = F_T(-\infty,\beta) - F_T(\alpha,\beta) = F_Q(-\infty,\beta,\alpha)$,

(iii) $F_T(\alpha,\beta) = F_Q(\alpha,\beta,\beta) = F_C(\beta,\beta) - F_C(\alpha,\beta)$.

Proof. Follows directly from corresponding disjoint partitions of the shapes and the additivity of the Lebesgue integral. □

Partition of Preisach plane into trapezoids, and primitive function Let either

$$(r_j)_{j\geq -1} = (-\infty, +\infty, m_0, M_0, m_1, M_1, \dots)$$

or

$$(r_j)_{j\geq -1} = (+\infty, -\infty, M_0, m_0, M_1, m_1, \dots)$$

be the prefixed reduced memory sequence of an input function u at time t. Setting $m_{-1} := -\infty$ and $M_{-1} := +\infty$, the special shape of the area $S_+(t)$ corresponding to (r_j) gives us the possibility of dividing $S_+(t)$ into trapezoids

$$Q_k := Q(m_k, M_k, m_{k+1}), \qquad k \geq -1,$$

or

$$Q_k := Q(m_k, M_{k+1}, m_{k+1}), \qquad k \geq -1,$$

respectively, i.e. those trapezoids with vertices

$$(m_k, m_k), (m_{k+1}, m_{k+1}), (m_{k+1}, M_k), (m_k, M_k)$$

or

$$(m_k, m_k), (m_{k+1}, m_{k+1}), (m_{k+1}, M_{k+1}), (m_k, M_{k+1}),$$

respectively, such that

$$S_+(t) = \dot{\bigcup}_{k\geq -1} Q_k$$

(see figure 2.12). Here, the left-most trapezoid Q_{-1} is the only unbounded one, and the right-most trapezoid (in the case where (r_j) is finite) may be given by

$$Q_{k'} := Q(m_{k'}, M_{k'}, M_{k'})$$

or

$$Q_{k'-1} := Q(m_{k'-1}, M_{k'}, M_{k'}),$$

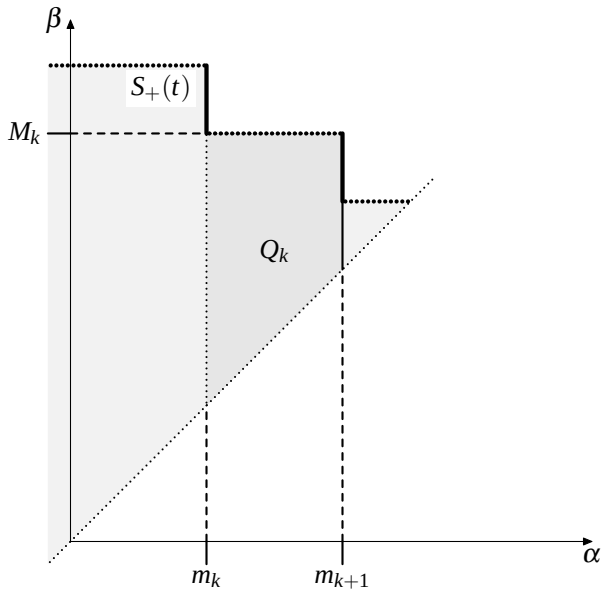


Figure 2.12: One trapezoid given by m_k, M_k, m_{k+1}

respectively, where k' is the highest index appearing among the M_k and m_k. Especially for the integral function $F(S_+(t)) = \int_{S_+(t)} d\mu$ it follows that

$$F(S_+(t)) = \sum_{k=-1}^{k'} F(Q_k),$$

or

$$F(S_+(t)) = \sum_{k=-1}^{k'-1} F(Q_k),$$

respectively, with

$$F(Q_k) = F_Q(m_k, M_k, m_{k+1})$$

or

$$F(Q_k) = F_Q(m_k, M_{k+1}, m_{k+1}),$$

respectively.

Principally, if the primitive function F_Q is known, it is easy to compute $F(S_+(t))$ and thus the Preisach operator Γ^μ at time t without integration: One just needs to add the several values $F(Q_k)$ which are direct applications of the primitive function F_Q. But this primitive function F_Q has the disadvantage to need three parameters, and one has to take special care of the rightmost trapezoid. We presented two other primitive functions which need only two parameters, F_C and F_T. We will show that they lead to even simpler formulas. But before doing this we compute primitive functions in some concrete examples. It is of course enough to know one of them, and we give it only for the triangles $T(\alpha, \beta)$.

Examples: (1) (Discrete parallel superposition)

We consider again the discrete superposition $\sum_{i=1}^{n} \omega_i \cdot (\Gamma^{\alpha_i,\beta_i} u)$ given by $(\alpha_i, \beta_i) \in \mathscr{P}$ for $i = 1, \ldots, n$ with $n \in \mathbb{N}$, and weights $\omega_i \in \mathbb{R}$. The primitive function F_T is then given by

$$F_T(\alpha, \beta) = \sum_{i=1}^{n} \omega_i \mathbf{1}_{T(\alpha,\beta)}(\alpha_i, \beta_i)$$

with the characteristic function

$$\mathbf{1}_{T(\alpha,\beta)}(\tilde{\alpha}, \tilde{\beta}) := \begin{cases} 1, & \text{if } (\tilde{\alpha}, \tilde{\beta}) \in T(\alpha, \beta), \\ 0, & \text{else.} \end{cases}$$

Taking into account the definition of the triangle $T(\alpha, \beta)$, this equation reduces to

$$F_T(\alpha, \beta) = \sum_{i=1}^{n} \tilde{\omega}_i$$

with

$$\tilde{\omega}_i = \begin{cases} \omega_i, & \text{if } \alpha_i < \alpha \text{ and } \beta_i < \beta, \\ 0, & \text{else} \end{cases}$$

for $i = 1 \ldots, n$.

In the special case $n = 1$ and $\omega_1 = 1$, i.e. in the case of the relay operator

$$(\Gamma^{\mu} u)(t) = (\Gamma^{\alpha_1 \beta_1} u)(t),$$

we just get

$$F_T(\alpha, \beta) = \begin{cases} 1, & \text{if } \alpha < \alpha_1 \text{ and } \beta_1 < \beta, \\ 0, & \text{else.} \end{cases}$$

(2) (Continuous parallel superposition)

Let the measure μ be given by the density function $\omega : \mathscr{P} \to \mathbb{R}$, i.e.

$$d\mu = \omega d\lambda$$

with

$$(\Gamma^{\mu} u)(t) = \iint\limits_{(\alpha,\beta) \in \mathscr{P}} \omega(\alpha, \beta)(\Gamma^{\alpha,\beta} u)(t) d\beta d\alpha.$$

The function $F_T(\alpha, \beta)$ is then given by

$$F_T(\alpha, \beta) = \iint\limits_{(\tilde{\alpha},\tilde{\beta}) \in T(\alpha,\beta)} \omega(\tilde{\alpha}, \tilde{\beta}) d\tilde{\beta} d\tilde{\alpha} = \int_{\alpha}^{\beta} \int_{\tilde{\alpha}}^{\beta} \omega(\tilde{\alpha}, \tilde{\beta}) d\tilde{\beta} d\tilde{\alpha}.$$

For $\omega \equiv 1$ on the triangle $T(A,B)$ and $\omega \equiv 0$ elsewhere, we get for example:

$$F_T(\alpha,\beta) = \int_{\alpha}^{\beta} \int_{\tilde{\alpha}}^{\beta} 1 d\tilde{\beta} d\tilde{\alpha} = \frac{1}{2}\alpha^2 - \alpha\beta + \frac{1}{2}\beta^2 = \frac{1}{2}(\alpha-\beta)^2$$

if $(\alpha,\beta) \in T(A,B)$. If $\alpha < A$ or $\beta > B$, we have to replace α by A, or β by B, respectively. We see that F_T is a piecewise quadratic polynomial in (α,β).

More complex examples of hysteresis can now easily be constructed through F_T. Let for example F_T be directly given by a finite number of overlapped second order polynomials in α and β:

$$F_T(\alpha,\beta) := \sum_{k=1}^{N} w_k(\alpha,\beta)\eta_k(\alpha,\beta)$$

with "local models"

$$\eta_k(\alpha,\beta) := \eta_k(\alpha,\beta;\theta_k^{\eta}) = a_k\alpha^2 + b_k\alpha\beta + c_k\beta^2 + d_k\alpha + e_k\beta + f_k$$

where $\theta_k^{\eta} := (a_k, b_k, c_k, d_k, e_k, f_k) \in \mathbb{R}^6$ are linear parameters, and "weights"

$$w_k(\alpha,\beta) := w_k(\alpha,\beta;\theta_k^{w})$$

where θ_k^{w} are nonlinear parameters. We give some examples:

Examples: **(1)** For $\mu = \lambda$ (i.e. $\omega \equiv 1$) we get the model of the second example above when $N = 1$ and

$$a_1 = 1/2, \qquad b_1 = -1, \qquad c_1 = 1/2, \qquad d_1 = e_1 = f_1 = 0,$$

if we consider only values (α,β) in the triangle $T(A,B)$.

(2) We define a second hysteresis model by

$$a_2 = 1/2, \qquad b_2 = -1/2, \qquad c_2 = 0, \qquad d_2 = e_2 = f_2 = 0.$$

With suitable definitions of weight functions, we can partially overlap the two hystereses, see figure 2.13.

(3) If we use "sharp" weight functions (for example the decision tree based weight functions resulting from $\sigma \to 0$) (see again chapter 1), we are able to reproduce exactly the hysteresis curves given by a mixture of Dirac measures $\mu = \sum_{i=1}^{n} \omega_i \delta_{(\alpha_i,\beta_i)}$ with

$$\delta_{(\alpha_i,\beta_i)}(B) = \begin{cases} 1, & \text{if } (\alpha_i,\beta_i) \in B, \\ 0, & \text{else,} \end{cases}$$

for $(\alpha_i,\beta_i) \in \mathscr{P}$ and $\omega_i \in \mathbb{R}$. We saw that in this case, we are able to define F_T as

$$F_T(\alpha,\beta) = \sum_{i=1}^{n} \tilde{\omega}_i$$

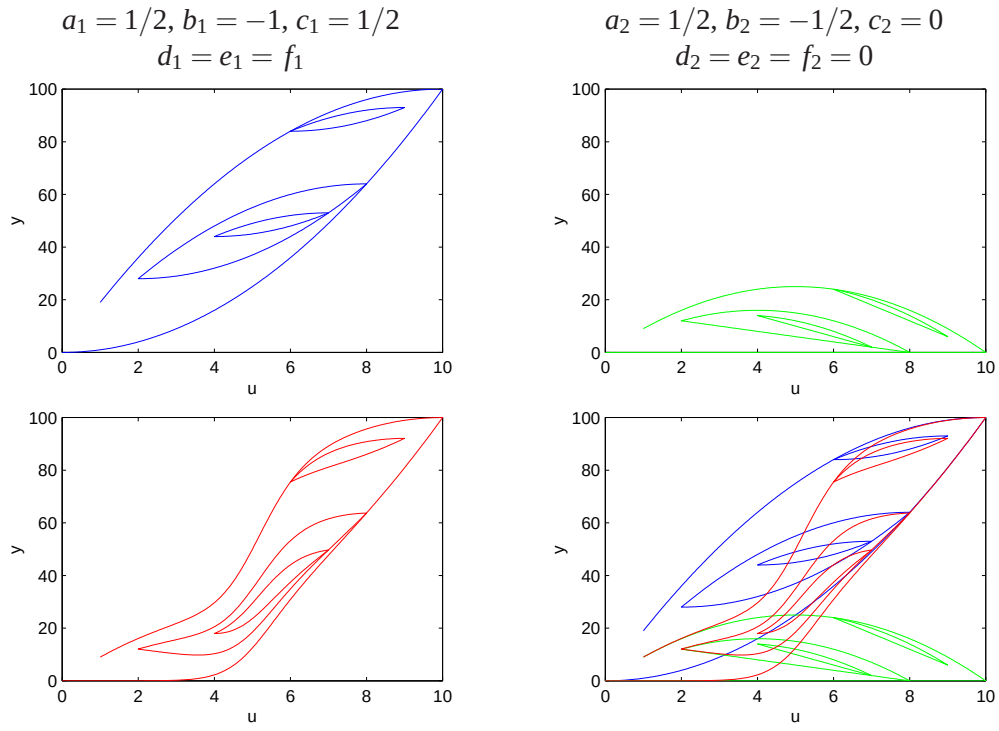


Figure 2.13: *u-y* diagram of a hysteresis. Upper row: Left: With density $\omega_1 \equiv 1$, i.e. $a_1 = 1/2$, $b_1 = -1$, $c_1 = 1/2$. Right: With parameters $a_2 = 1/2$, $b_2 = -1/2$, $c_2 = 0$. Lower Row: Left: Two overlapped local models. Right: All three hysteresis curves in one picture.

with

$$\tilde{\omega}_i = \begin{cases} \omega_i, & \text{if } \alpha < \alpha_i \text{ and } \beta_i < \beta, \\ 0, & \text{else} \end{cases}$$

for $i = 1 \ldots, n$. It is easily seen that the function F_T partitions the Preisach plane into parts with boundaries parallel to one of the axes. Such a partition is easily done by a decision tree used to construct the weight functions. In the simplest case $\Gamma^{\mu} = \Gamma^{\alpha_1,\beta_1}$, i.e. in the case of the relay operator, we saw that

$$F_T(\alpha,\beta) = \begin{cases} 1, & \text{if } \alpha < \alpha_1 \text{ and } \beta_1 < \beta, \\ 0, & \text{else.} \end{cases}$$

The model η is then given by

$$\eta(\alpha,\beta) = \sum_{j=1}^{3} w_j(\alpha,\beta)\eta_j(\alpha,\beta)$$

with

$$\begin{aligned} w_1(\alpha,\beta) &= 1 - \mathbf{1}_{\alpha<\alpha_1}(\alpha,\beta), & \eta_1(\alpha,\beta) &= 0, \\ w_2(\alpha,\beta) &= \mathbf{1}_{\alpha<\alpha_1}(\alpha,\beta)\mathbf{1}_{\beta\le\beta_1}(\alpha,\beta), & \eta_2(\alpha,\beta) &= 0, \\ w_3(\alpha,\beta) &= \mathbf{1}_{\alpha<\alpha_1}(\alpha,\beta)(1 - \mathbf{1}_{\beta\le\beta_1}(\alpha,\beta)), & \eta_3(\alpha,\beta) &= 1. \end{aligned}$$

Antisymmetric extension and summation formula Given the primitive functions F_Q, F_C, and F_T, it is easy to implement the hysteresis operator as a computer programme. To have simpler formulas, we consider a natural extension of these functions:

Let F be any function defined only for (α,β) with $-\infty \le \alpha < \beta \le +\infty$, e.g. $F = F_Q$, F_C, or F_T. We extend this function to its ***antisymmetric extension*** (also denoted by F) to the whole $\overline{\mathbb{R}}^2$, defining:

$$F(\alpha,\beta) := -F(\beta,\alpha) \qquad \text{if } \alpha > \beta,$$

as well as

$$F(\alpha,\beta) := 0 \qquad \text{if } \alpha = \beta.$$

Theorem 2.4: *Let μ be a Preisach measure and let F_T be the antisymmetric extension of the primitive function with respect to the triangles $T(\alpha,\beta)$. Let further*

$$u \in C^0(\mathscr{T}_{t_0})$$

be an input function and (r_j) be the prefixed reduced memory sequence at a time $t \in \mathscr{T}_{t_0}$ with either

$$(r_{-1}, r_0) = (-\infty, +\infty) \qquad \textit{or} \qquad (r_{-1}, r_0) = (+\infty, -\infty).$$

Let

$$j_0 := \begin{cases} -1, & \textit{if } (r_{-1}, r_0) = (-\infty, +\infty), \\ 0, & \textit{if } (r_{-1}, r_0) = (+\infty, -\infty), \end{cases}$$

and let $m \in \mathbb{N} \cup \{\infty\}$ be such that $m+2$ denotes the length of the sequence (r_j). Under the assumption that

$$S_+(t_0) = \begin{cases} \mathscr{P}, & \text{if } (r_{-1}, r_0) = (-\infty, +\infty), \\ \emptyset, & \text{if } (r_{-1}, r_0) = (+\infty, -\infty), \end{cases}$$

we have the summation formula

$$F(S_+(t)) = \sum_{j=j_0}^{m-1} F_T(r_j, r_{j+1}).$$

(The summation begins thus always with that coefficient r_j which is equal to $-\infty$; empty sums are considered to be equal to zero).

Proof. Since we know that under the given assumptions $S_+(t) = S_+((r_j))$ is valid where (r_j) is the prefixed reduced memory sequence of u at time t, it is enough to show that the formulas in the theorem are correct for $S_+((r_j))$ instead of $S_+(t)$.

We consider first the case where (r_j) is finite and use induction over the length $m+2$ of (r_j). Let first $m = 0$. Then, it holds that $S_+((r_j)) = S_+(t_0)$. If $S_+(t_0) = \mathscr{P}$, we have

$$(r_j) = (-\infty, +\infty)$$

and thus

$$\sum_{j=-1}^{-1} F_T(r_j, r_{j+1}) = F_T(-\infty, +\infty) = F(\mathscr{P}) = F\big(S_+((r_j))\big).$$

If $S_+(t_0) = \emptyset$, we have

$$(r_j) = (+\infty, -\infty)$$

and thus

$$\sum_{j=0}^{-1} F_T(r_j, r_{j+1}) = 0 = F(\emptyset) = F\big(S_+((r_j))\big).$$

To go inductively from m to $m+1$, we consider $S_+((r_j)_{j=-1,\dots,m})$. This is the polyhedron with vertices $(-\infty, -\infty)$, (r_j, r_{j+1}) for $j = j_0, \dots, m$ and (r_m, r_m), where $j_0 = -1$ or $j_0 = 0$. To get the polyhedron for the step $m+1$, $S_+((r_j)_{j=-1,\dots,m+1})$, one has to remove the vertex (r_m, r_m) and to replace it by the vertices (r_m, r_{m+1}) and (r_{m+1}, r_{m+1}). This is the same as adding or removing a triangle:

- If $r_m < r_{m+1} < r_{m-1}$, then one appends the triangle $T(r_m, r_{m+1})$ to $S_+((r_j)_{j=-1,\dots,m})$ to get $S_+((r_j)_{j=-1,\dots,m+1})$, thus

$$\begin{aligned} F(S_+((r_j)_{j=-1,\dots,m+1})) &= F(S_+((r_j)_{j=-1,\dots,m})) + F(T(r_m, r_{m+1})) \\ &= \sum_{j=j_0}^{m-1} F_T(r_j, r_{j+1}) + F_T(r_m, r_{m+1}). \end{aligned}$$

- If $r_{m-1} < r_{m+1} < r_m$, then one subtracts the triangle $T(r_{m+1}, r_m)$ from $S_+((r_j)_{j=-1,\dots,m})$ to get $S_+((r_j)_{j=-1,\dots,m+1})$, thus

$$\begin{aligned} F(S_+((r_j)_{j=-1,\dots,m+1})) &= F(S_+((r_j)_{j=-1,\dots,m})) - F(T(r_{m+1}, r_m)) \\ &= \sum_{j=j_0}^{m-1} F_T(r_j, r_{j+1}) + F_T(r_m, r_{m+1}). \end{aligned}$$

In the infinite case, we first assume that the measure μ is non-negative. Then, for all $j \geq j_0$, $F_T(r_j, r_{j+1})$ is ≥ 0 if $r_j < r_{j+1}$ and ≤ 0 else. Furthermore, because of the alternating property, i.e.

$$r_j < r_{j+2} < r_{j+1} \qquad \text{or} \qquad r_{j+1} < r_{j+2} < r_j,$$

we have that

$$T(r_j, r_{j+1}) \supsetneqq T(r_{j+2}, r_{j+1}) \qquad \text{or} \qquad T(r_{j+1}, r_j) \supsetneqq T(r_{j+1}, r_{j+2}),$$

respectively, and thus in all cases

$$|F_T(r_j, r_{j+1})| > |F_T(r_{j+1}, r_{j+2})|$$

using the antisymmetry of F_T. Since

$$T(r_{j_0}, r_{j_0+1}) \supsetneqq T(r_{j_0+2}, r_{j_0+1}) \supsetneqq T(r_{j_0+2}, r_{j_0+3}) \supsetneqq T(r_{j_0+4}, r_{j_0+3}) \supsetneqq \dots$$

where the intersection over all these sets is $T(r_\infty, r_\infty) = \emptyset$ with $r_\infty := \lim_{j\to\infty} r_j$, the sequence

$$|F_T(r_j, r_{j+1})|, \qquad j \geq j_0,$$

converges to 0 (this follows from the continuity from above of the measure μ).

In conclusion, $(F_T(r_j, r_{j+1}))$ is an alternating sequence with strictly decreasing absolute values which converges to 0. From this, we have also that the series $\sum_{j=j_0}^{\infty} F_T(r_j, r_{j+1})$ converges.

After having proved that the series converges at all, we have to show that it converges to the right value. Since

$$S_+((r_{j_0})) \subsetneqq S_+((r_{j_0}, r_{j_0+1}, r_{j_0+2})) \subsetneqq \dots \subsetneqq S_+((r_j)_{j=j_0,\dots,j_0+2k}) \subsetneqq \dots$$

and the union of all these sets is $S_+((r_j)_{j_0 \geq 0})$, we have that the partial sequence

$$\left(\sum_{j=j_0}^{j_0} F_T(r_j, r_{j+1}), \sum_{j=j_0}^{j_0+2} F_T(r_j, r_{j+1}), \dots, \sum_{j=j_0}^{j_0+2k} F_T(r_j, r_{j+1}), \dots \right)$$

converges to $F(S_+((r_j)_{j_0 \geq 0}))$, and thus the complete sequence, that is the series

$$\sum_{j=j_0}^{\infty} F_T(r_j, r_{j+1}),$$

must also converge to this value (after having shown that it converges at all).

If μ is signed, the result follows immediately from the decomposition

$$\mu = \mu_+ - \mu_-$$

with non-negative measures μ_+ and μ_-. □

Corollary: *With the same assumptions as in the theorem, and if the sequence (r_j) is finite of length $m+2$, we have for the antisymmetric extended primitive functions F_C and F_Q:*

$$F(S_+(t)) = -\sum_{j=j_0}^{m-1} F_C(r_j, r_{j+1}) + \begin{cases} F_C(r_m, r_m), & \text{if } j_0 \equiv m-1 \bmod 2, \\ 0, & \text{else,} \end{cases}$$

and

$$F(S_+(t)) = \sum_{\substack{j=j_0+2k \\ k\in\mathbb{N}}}^{m-2} F_Q(r_j, r_{j+1}, r_{j+2}) + \begin{cases} F_Q(r_{m-1}, r_m, r_m), & \text{if } j_0 \equiv m-1 \bmod 2, \\ 0, & \text{else.} \end{cases}$$

If (r_j) is infinite, the infinite series $\sum_{j=j_0}^{\infty} F_C(r_j, r_{j+1})$ does not convergence. In contrast, for F_Q, we have

$$F(S_+(t)) = \sum_{\substack{j=j_0+2k \\ k\in\mathbb{N}}}^{\infty} F_Q(r_j, r_{j+1}, r_{j+2}).$$

Proof. Let (r_j) be of length $m+2$. For $\alpha \le \beta$, we have

$$F_T(\alpha, \beta) = F_C(\beta, \beta) - F_C(\alpha, \beta).$$

Thus:

$$\begin{aligned} F(S_+(t)) = \sum_{j=j_0}^{m-1} F_T(r_j, r_{j+1}) &= \sum_{\substack{j=j_0+2k \\ k\in\mathbb{N}}}^{m-1} F_T(r_j, r_{j+1}) - \sum_{\substack{j=j_0+2k+1 \\ k\in\mathbb{N}}}^{m-1} F_T(r_{j+1}, r_j) \\ &= \sum_{\substack{j=j_0+2k \\ k\in\mathbb{N}}}^{m-1} F_C(r_{j+1}, r_{j+1}) - \sum_{\substack{j=j_0+2k \\ k\in\mathbb{N}}}^{m-1} F_C(r_j, r_{j+1}) \\ &\quad - \sum_{\substack{j=j_0+2k+1 \\ k\in\mathbb{N}}}^{m-1} F_C(r_j, r_j) + \sum_{\substack{j=j_0+2k+1 \\ k\in\mathbb{N}}}^{m-1} F_C(r_{j+1}, r_j). \end{aligned}$$

Shifting the index j to $j+1$ in the third sum and combining it with the first sum, as well as combining the second and forth sums leads to the first statement of the corollary.

For $\alpha \le \beta$, we also have

$$F_T(\alpha, \beta) = F_Q(\alpha, \beta, \beta).$$

Thus:

$$F(S_+(t)) = \sum_{j=j_0}^{m-1} F_T(r_j, r_{j+1}) = \sum_{\substack{j=j_0+2k \\ k\in\mathbb{N}}}^{m-1} F_T(r_j, r_{j+1}) - \sum_{\substack{j=j_0+2k+1 \\ k\in\mathbb{N}}}^{m-1} F_T(r_{j+1}, r_j)$$
$$= \sum_{\substack{j=j_0+2k \\ k\in\mathbb{N}}}^{m-1} F_Q(r_j, r_{j+1}, r_{j+1}) - \sum_{\substack{j=j_0+2k+1 \\ k\in\mathbb{N}}}^{m-1} F_Q(r_{j+1}, r_j, r_j).$$

Shifting the index j to $j+1$ in the last sum and mixing the two resulting sums, we get

$$F(S_+(t)) = \sum_{\substack{j=j_0+2k \\ k\in\mathbb{N}}}^{m-2} [F_Q(r_j, r_{j+1}, r_{j+1}) - F_Q(r_{j+2}, r_{j+1}, r_{j+1})]$$
$$+ \begin{cases} F_Q(r_{m-1}, r_m, r_m), & \text{if } j_0 \equiv m-1 \bmod 2, \\ 0, & \text{else.} \end{cases}$$

Since

$$F_Q(r_j, r_{j+1}, r_{j+1}) - F_Q(r_{j+2}, r_{j+1}, r_{j+1}) = F_Q(r_j, r_{j+1}, r_{j+2})$$

we get the second statement of the corollary.

In the infinite case, the reason that $\sum_{j=j_0}^{\infty} F_C(r_j, r_{j+1})$ diverges is that $F_C(r_j, r_{j+1})$ does in general not converge to zero as $j \to \infty$: it converges to $F_C(r_\infty, r_\infty)$ with

$$r_\infty := \lim_j r_j.$$

In the case of F_Q we saw that the

$$Q(r_j, r_{j+1}, r_{j+2}), \qquad j = j_0 + 2k, k \in \mathbb{N}$$

constitute a disjoint partition of $S_+(t)$, and the result follows from the σ-additivity of μ. □

Remark: There are similar (dual) formulas for the complement area $S_-(t)$ indicating the hysterons which are negatively saturated. We just give the one for F_T. From the fact that

$$\mathscr{P} = S_+(t) \dot\cup S_-(t)$$

we get

$$\begin{aligned}
F(S_-(t)) &= F(\mathscr{P}) - F(S_+(t)) \\
&= F_T(-\infty,+\infty) - \sum_{j=j_0}^{m-1} F_T(r_j, r_{j+1}) \\
&= F_T(-\infty,+\infty) - \left\{ \begin{array}{ll} F_T(r_{-1},r_0), & \text{if } (r_{-1},r_0) = (-\infty,+\infty), \\ 0, & \text{if } (r_{-1},r_0) = (+\infty,-\infty), \end{array} \right\} - \sum_{j=0}^{m-1} F_T(r_j, r_{j+1}) \\
&= \left\{ \begin{array}{ll} 0, & \text{if } (r_{-1},r_0) = (-\infty,+\infty), \\ -F_T(+\infty,-\infty), & \text{if } (r_{-1},r_0) = (+\infty,-\infty), \end{array} \right\} - \sum_{j=0}^{m-1} F_T(r_j, r_{j+1}) \\
&= - \sum_{j=1-j_0}^{m-1} F_T(r_j, r_{j+1}).
\end{aligned}$$

Thus, apart from the sign, the only difference with respect to $F(S_+(t))$ is the different starting point $1-j_0$ instead of j_0. The first term in the sum corresponds to the coefficient in (r_j) which has the value $+\infty$.

Remark: The summation formulas given in the last theorem are not new. The formula involving F_T can be found e.g. in Mayergoyz [1991], but we extended it to the infinite case as well as to the remaining primitive shapes. To the author's knowledge, also the use of the prefixed reduced memory sequences is new.

Computation of the Preisach operator We are now able to give a simple formula for the computation of the Preisach operator Γ^μ with input function $u \in C^0(\mathscr{T}_{t_0})$ and the initial condition that $S_+(t_0)$ is a polyhedron such that the common boundary $A(t_0)$ of $S_+(t_0)$ and $S_-(t_0)$ is a polygon, and a decreasing graph with only axis parallel edges. Then, for a given time t, we have the formula

$$(\Gamma^\mu u)(t) = 2 \int\limits_{S_+(t)} d\mu - \int\limits_{\mathscr{P}} d\mu = F(S_+(t)) - F(\mathscr{P}),$$

with

$$F(S_+(t)) := \int\limits_{S_+(t)} d\mu \qquad \text{and} \qquad F(\mathscr{P}) := \int\limits_{\mathscr{P}} d\mu.$$

We first have to compute the prefixed reduced memory sequence (r_j) corresponding to $S_+(t_0)$, u and t. This can be done as follows: According to the assumptions, $S_+(t_0)$ corresponds to a prefixed reduced memory sequence (r'_j). The coefficients r'_j may be thought as certain minima and maxima which occurred in the input function u before time t_0. Therefore we could "prefix" these values to the function $u \in C^0(\mathscr{T}_{t_0})$, for example by defining

$$u(t_0 - 1 + 2^{-j}) := r_j, \qquad j = -1, 0, \ldots$$

and interpolating monotonously. To compute the prefixed reduced memory sequence (r_j) for a time $t \geq t_0$, one can use the algorithm given for the definition of the reduced memory sequence, applied to the augmented input function u, even though the values $-\infty$ and $+\infty$ occur and u may have a jump at t_0. For discrete settings, we will describe in the next paragraph a recursive procedure which is more practicable.

Having got (r_j), since

$$F(S_+(t)) = \sum_{j=j_0}^{m-1} F_T(r_j, r_{j+1})$$

and

$$F(\mathscr{P}) = F_T(-\infty, +\infty),$$

we just have to compute

$$(\Gamma^\mu u)(t) = 2 \sum_{j=j_0}^{m-1} F_T(r_j, r_{j+1}) - F_T(-\infty, +\infty)$$

where

$$j_0 := \begin{cases} -1, & \text{if } (r_{-1}, r_0) = (-\infty, +\infty), \\ 0, & \text{if } (r_{-1}, r_0) = (+\infty, -\infty). \end{cases}$$

Recursive computation of reduced memory sequences We have to compute the reduced memory sequence for an input $u(t)$ at several times t. Practically, one works in a discrete setting. Thus, the time steps are given by discrete values

$$t_0, t_0 + 1, t_0 + 2, \ldots$$

or more general, with non-equidistant time steps

$$t_0 < t_1 < t_2 < \cdots$$

which may be handled in exactly the same way. Principally, for the computation of the prefixed reduced memory sequence at each time step, the algorithm given as the definition for the prefixed reduced memory sequence can be used. But it is better to use a recursive procedure which updates the prefixed reduced memory sequence at time t to the prefixed reduced memory sequence at time $t + 1$, than to compute it each time from the scratch.

Remark: In the discrete setting, we want to adopt the following convention: We assume that at time t, the system is in the state $x(t)$ and we apply the input (control) $u(t)$. Then the system will run until time $t + 1$, changing its state to $x(t + 1)$, and producing the output $y(t + 1)$ in accordance with the axioms of dynamical systems in section 2.1. We are thus in the following situation: Given state $x(t)$ and input $u(t)$, we will get state $x(t + 1)$ and output $y(t + 1)$ in the next time step. The state in the case of Preisach hysteresis is given by the prefixed reduced memory sequence:

$$x(t) = (r_j)(t).$$

The following algorithm is applicable: Assume that a prefixed reduced memory sequence $(r_j)_{j=-1,\dots,m} = (r_j(t))_{j=-1,\dots,m(t)}$ at time t for the input sequence $u(\cdot)$ is given. We want to compute the new reduced memory sequence $(r'_j)_{j=-1,\dots,m'}$ for the time $t+1$. The only new additional datum is $u(t)$. Set $J = m$ and check the following:

- If $r_J = u(t)$ then set $r'_j := r_j$ for $j = 1,\dots,J$ and $m' := J$, and we are done.
- Else, if $r_{J-1} < u(t) < r_J$ or $r_{J-1} > u(t) > r_J$, then set $r'_j := r_j$ for $j = 1,\dots,J$, $r'_{J+1} = u(t)$ and $m' := J+1$, and we are done.
- Else, set $J \leftarrow J-1$ and repeat the procedure.

Since $u(t) \in \mathbb{R}$, the algorithm stops at latest when checking the above conditions for $J = 0$: We have either that (r_{-1}, r_0) equals $(-\infty, +\infty)$ and then $r_{-1} < u(t) < r_0$, or we have that (r_{-1}, r_0) equals $(+\infty, -\infty)$ and then $r_{-1} > u(t) > r_0$ holds. The algorithm reduces step by step the original prefixed reduced memory sequence by cutting coefficients from the tail. Only in the last step, the additional coefficient $u(t)$ may be appended.

We remark also that the prefixed reduced memory sequence has always at least length 3 with the only possible exception of the initial state at time t_0, because for every input $u \in \mathbb{R}$, one has $-\infty < u(t) < +\infty$, with the consequence that $u(t)$ is always the last coefficient of the sequence.

Complete computation For a given primitive function F_T corresponding to a Preisach measure μ, an input sequence u, and an initial prefixed reduced memory sequence

$$x_0 = x(t_0) := (r_j)(t_0)$$

at the initial time t_0, we compute recursively for $t = t_0, t_0+1, t_0+2, \dots$:

- the state $x(t+1) = (r_j)(t+1)$ using the recursive algorithm from $(r_j)(t)$ and $u(t)$, and
- the output $y(t+1)$ according to theorem 2.4 (setting $(r_j)_{j=-1,\dots,m} := (r_j)_{j=-1,\dots,m}(t+1)$):

$$y(t+1) := (\Gamma^\mu u)(t+1) = 2\sum_{j=j_0}^{m-1} F_T(r_j, r_{j+1}) - F_T(-\infty, +\infty)$$

where

$$j_0 := \begin{cases} -1, & \text{if } (r_{-1}, r_0) = (-\infty, +\infty), \\ 0, & \text{if } (r_{-1}, r_0) = (+\infty, -\infty). \end{cases}$$

Generalizations of the Preisach model

In the following we shortly consider some generalizations of the Preisach model.

Affine-linear transformations of the Preisach model It should be noted that the choice of the output of the hysterons to be in the set $\{-1,+1\}$ is rather arbitrary. We could easily replace these values by any other values $\{y_-,y_+\}$ as long as $y_- < y_+$ and build the Preisach model with the obtained hysterons. Using the affine-linear transformation

$$\psi(y) = ay + b$$

with $a := 1/2(y_+ - y_-)$ and $b := 1/2(y_+ + y_-)$ the transformation of the hysterons and the Preisach model can easily be described. We get for the "transformed" Preisach model

$$\begin{aligned} y(t) := (\Gamma^{\mu,\psi}_{t_0,x_0}u)(t) &= \int_{(\alpha,\beta)\in\mathscr{P}} (\Gamma^{\alpha,\beta,\psi}_{t_0,x_0(\alpha,\beta)}u)(t)d\mu = \int_{(\alpha,\beta)\in\mathscr{P}} \psi((\Gamma^{\alpha,\beta}_{t_0,x_0(\alpha,\beta)}u)(t))d\mu \\ &= a\int_{(\alpha,\beta)\in\mathscr{P}} (\Gamma^{\alpha,\beta}_{t_0,x_0(\alpha,\beta)}u)(t))d\mu + b\mu(\mathscr{P}) = a(\Gamma^{\mu}_{t_0,x_0}u)(t) + b\mu(\mathscr{P}). \end{aligned}$$

Whereas the scaling constant a can be subsumed into the measure μ, the translation b really extends the model class of the Preisach family by adding a translational constant $b\mu(\mathscr{P})$. We get the following summation formula:

$$y(t) = a(\Gamma^{\mu}u)(t) + b\mu(\mathscr{P}) = \sum_{j=j_0}^{m-1} 2aF_T(r_j,r_{j+1}) + (b-a)F_T(-\infty,+\infty)$$

with

$$j_0 := \begin{cases} -1, & \text{if } (r_{-1},r_0) = (-\infty,+\infty), \\ 0, & \text{if } (r_{-1},r_0) = (+\infty,-\infty). \end{cases}$$

Considering that a and b can be chosen arbitrarily, this lets us redefine the primitive function in the following way to further simplify the summation formula:

$$F(\alpha,\beta) := \begin{cases} (a+b)F_T(-\infty,+\infty) & \text{if } (\alpha,\beta) = (-\infty,+\infty), \\ (a-b)F_T(-\infty,+\infty) & \text{if } (\alpha,\beta) = (+\infty,-\infty), \\ 2aF_T(\alpha,\beta) & \text{else.} \end{cases}$$

$F(\alpha,\beta)$ is still antisymmetric except for $(\alpha,\beta) = (-\infty,+\infty)$. The summation formula becomes then fairly easy:

$$y(t) = a(\Gamma^{\mu}u)(t) + b\mu(\mathscr{P}) = \sum_{j=0}^{m-1} F(r_j,r_{j+1}).$$

Relaxing the antisymmetry As a further generalization, we could also completely relax the constraint on the function F_T and thus on F to be antisymmetric. This results in hysteresis loops which are not closed when $F(\alpha,\beta) \neq -F(\beta,\alpha)$. And even if F is continuous and μ is absolutely continuous with respect to the Lebesgue measure, the relation $F(\alpha,\alpha) \neq 0$ leads to jumps on the turning points.

2.2.3 Identification

The Preisach model is a superposition of simple relay operators weighted by a measure μ. This μ is usually not known! In the *classical* Preisach model, the measure μ is given by a density function ω, also called Preisach function. There exist two common approaches of identification methods for the classical Preisach hysteresis model (with density function), the lookup table approach and the basis function approach, respectively (see e.g. Kirchmair [2002]). The lookup table approach uses a primitive function similar to the one we are using. The values of this function under a regular grid have to be estimated by a special identification method which requires prescribed input sequences. Points not on the grid are linearly interpolated. The second method approximates the Preisach function ω directly. In this case, it is written as

$$\omega(\alpha,\beta) = \sum_j a_j w_j(\alpha,\beta)$$

with constants a_j which must be estimated, and fixed weight functions w_j given by Gaussian bell functions. During the simulation of the Preisach model with these particular weight functions, the weight functions have to be integrated over triangular areas. This can only be done numerically, because no analytic solutions exist. Our approach is to some extend a combination and generalization of these two methods. In some way, we approximate the (unknown) primitive function F of the Preisach function ω (or more generally of the measure μ) by a parameterized function $\hat{F}(\alpha,\beta;\theta)$ where the parameter vector θ has to be identified by measured input/output data. Here, we use a variant of the LOLIMOT algorithm (we described the original algorithm already in chapter 1). Later, in chapter 5, we will propose another identification scheme which allows the treatment of more general models.

Long-time memory for LOLIMOT The original LOLIMOT algorithm uses local models of linear ARX type. These models do not provide a long-time memory. As the inputs in these models are only the last n_u values of the input u and the last n_y values of the output y backwards in time, where n_u and n_y are fixed natural numbers, this only provides some kind of short-time memory which reaches back to $\max\{n_u, n_y\}$ of time steps in the past. Therefore, this version of the LOLIMOT algorithm is not able to identify Preisach hysteresis. By approximating the primitive function F of the Preisach model we will be able to make the standard LOLIMOT algorithm apply to this kind of models, too.

Identification with the general summation formula For identification, it is best to use the general summation formula obtained in the last subsection:

$$y(t) = a(\Gamma^\mu u)(t) + b\mu(\mathscr{P}) = \sum_{j=0}^{m-1} F(r_j, r_{j+1}).$$

As mentioned, the representation sequence has always at least 3 entries, r_{-1}, r_0, and r_1, with possible exception of the initial sequence. But this latter sequence will never be used for calculation of an output through the summation formula. The summation formula has therefore always at least the two terms

$$F(r_{-1}, r_0) + F(r_0, r_1)$$

which is

$$F(+\infty,-\infty)+F(-\infty,r_1) \qquad \text{or} \qquad F(-\infty,+\infty)+F(+\infty,r_1),$$

such that in each case these two terms can (and should) be concatenated into one. We have to estimate the following "parameters":

- the 1-dimensional functions

  $$F_-(\beta) := F(+\infty,-\infty)+F(-\infty,\beta) \qquad \text{for all } \beta \in \mathbb{R},$$

  and

  $$F_+(\alpha) := F(-\infty,+\infty)+F(+\infty,\alpha) \qquad \text{for all } \alpha \in \mathbb{R},$$

  and

- the 2-dimensional function $F(\alpha,\beta)$ for $\alpha,\beta \in \mathbb{R}$ with $\alpha < \beta$ in the antisymmetric case and $\alpha \neq \beta$ in the general case.

The function $F_-(\beta) = F(+\infty,-\infty)+F(-\infty,\beta)$ is needed only for the case of initial negative saturation, i.e. $(r_{-1},r_0) = (+\infty,-\infty)$, and the function $F_+(\alpha) = F(-\infty,+\infty)+F(+\infty,\alpha)$ only in the case of initial positive saturation, i.e. $(r_{-1},r_0) = (-\infty,+\infty)$.

We therefore can write the summation formula as

$$y(t) = F_\pm(r_1) + \sum_{j=1}^{m-1} F(r_j,r_{j+1}).$$

Linearly parameterized models We begin by modelling F as a linerarly parameterized function:

$$F(\alpha,\beta) = \theta^\top \varphi(\alpha,\beta).$$

It makes sense to use separate parameters when α or β equals $-\infty$ or $+\infty$. If we define the regression vector as

$$\theta = (\theta_-^*, \theta_+^*, \theta^*)^\top,$$

then we want to understand $\theta^\top\varphi(\alpha,\beta)$ as

$$\theta^\top \varphi(\alpha,\beta) = \begin{cases} \theta^\top\varphi_-(\alpha,\beta) & \text{if } (r_{-1},r_0) = (+\infty,-\infty), \\ \theta^\top\varphi_+(\alpha,\beta) & \text{if } (r_{-1},r_0) = (-\infty,+\infty), \end{cases}$$

with

$$\varphi_-(\alpha,\beta) = (\varphi_-^*(\beta) \quad 0 \quad \varphi^*(\alpha,\beta))^\top$$

and

$$\varphi_+(\alpha,\beta) = (0 \quad \varphi_+^*(\alpha) \quad \varphi^*(\alpha,\beta))^\top$$

where $\varphi_-^*(\beta)$, $\varphi_+^*(\alpha)$ and $\varphi^*(\alpha,\beta)$ belong to the parameters θ_-^*, θ_+^*, and θ^* respectively, such that

$$F_-(\beta) = \theta_-^{*\top}\varphi_-^*(\beta), \qquad F_+(\alpha) = \theta_+^{*\top}\varphi_+^*(\alpha), \qquad \text{and} \qquad F(\alpha,\beta) = \theta^{*\top}\varphi^*(\alpha,\beta).$$

This is directly implementable, and e.g. linear regression can be used directly with the regressor matrix consisting of several vectors $\varphi_-(\alpha,\beta)$ and $\varphi_+(\alpha,\beta)$.

If the function $F(\alpha,\beta)$ shall be antisymmetric, one should split $\varphi^*(\alpha,\beta)$ into

$$\varphi^*(\alpha,\beta) = \begin{cases} \varphi^*(\alpha,\beta), & \text{if } \alpha < \beta, \\ -\varphi^*(\beta,\alpha), & \text{if } \alpha > \beta. \end{cases}$$

Examples: (a) The simplest choice is obviously the affine-linear model

$$F_-(\beta) := b_-\beta + c_-, \qquad F_+(\alpha) := a_+\alpha + c_+, \qquad \text{and} \qquad F(\alpha,\beta) := a\alpha + b\beta + c,$$

i.e.

$$\varphi_-^*(\beta) := (\beta,1)^\top, \qquad \varphi_+^*(\alpha) := (\alpha,1)^\top, \qquad \text{and} \qquad \varphi^*(\alpha,\beta) := (\alpha,\beta,1)^\top$$

and

$$\theta_-^* := (b_-,c_-)^\top \in \mathbb{R}^2, \qquad \theta_+^* := (a_+,c_+)^\top \in \mathbb{R}^2, \qquad \text{and} \qquad \theta^* := (a,b,c)^\top \in \mathbb{R}^3.$$

(b) A generalization is given by higher order polynomials; for the second order polynomial, we have

$$F_-(\beta) := c_-\beta^2 + e_-\beta + f_-, \qquad F_+(\alpha) := a_+\alpha^2 + d_+\alpha + f_+$$

and

$$F(\alpha,\beta) := a\alpha^2 + b\alpha\beta + c\beta^2 + d\alpha + e\beta + f$$

i.e.

$$\varphi^*(\alpha,\beta) := (\alpha^2,\alpha\beta,\beta^2,\alpha,\beta,1)^\top \in \mathbb{R}^6$$

and

$$\theta^* := (a,b,c,d,e,f)^\top \in \mathbb{R}^6,$$

and similarly for φ_-^*, θ_-^* as well as φ_+^*, θ_+^*.

Interpretation of F Other linearly parameterized functions are possible. The decision what choice should be taken is surely not easy, but it can be remarked that the curves between the turning points of the hysteresis loops in the *u-y* diagram are congruent in some sense to the graph of the function F: A look on the summation formula shows that the first curve of the outer loop is given by either $y(t) = F_-(u(t))$ or $y(t) = F_+(u(t))$. Thus we get a picture of the graph of either $F_-(u(t))$ or $F_+(u(t))$. After the first turning point $u(t_1)$ we have

$$y(t) = F_-(u(t_1)) + F(u(t_1),u(t)) \qquad \text{or} \qquad y(t) = F_+(u(t_1)) + F(u(t_1),u(t))$$

and we have got a translated graph of F where one component is fixed. And so on.

In praxis, occurring hysteresis curves often show a sigmoid shape because of saturation effects. This kind of curves cannot be well approximated with the above examples. The usual sigmoid function could be used but contain parameters which are not linearly parameterized. Nevertheless, one could try to model this kind of hysteresis by weighted superpositions of linearly parameterized functions: the idea of local models may help here.

"Local models" We consider superpositions of weighted local models,

$$F(\alpha,\beta) = \sum_{k=1}^{N} w_k(\alpha,\beta)\eta_k(\alpha,\beta),$$

and similar for $F_-(\beta)$ and $F_+(\alpha)$. The η_k may be linearly parameterized:

$$\eta_k(\alpha,\beta) = \theta_k^{\eta\top}\varphi(\alpha,\beta).$$

It should be noted that due to the linearity of the summation formula, the parameters θ_k^{η} remain linear in the global model provided the weights $w_k(\alpha,\beta)$ are assumed to be known:

$$(\Gamma^{\mu}u)(t) = \sum_{j=0}^{m(t)-1} F(r_j(t), r_{j+1}(t)) = \sum_{j=0}^{m(t)-1}\sum_{k=1}^{N} w_k(r_j(t), r_{j+1}(t))\theta_k^{\eta\top}\varphi(r_j(t), r_{j+1}(t)).$$

This opens the door to the usage of the LOLIMOT algorithm.

Identification with the LOLIMOT algorithm If we take

$$\eta^{(N)}(\alpha,\beta) := \sum_{k=1}^{N} w_k(\alpha,\beta)\theta_k^{\eta\top}\varphi(\alpha,\beta)$$

as an approximation for $F(\alpha,\beta)$, then the whole hysteresis model Γ^{μ} is approximated by

$$(\Gamma^{\mu}u)(t) \approx \sum_{j=0}^{m(t)-1} \eta^{(N)}(r_j(t), r_{j+1}(t)).$$

As in the original LOLIMOT algorithm, we can construct the weights

$$w_k(\alpha,\beta) := w_k(\alpha,\beta;\theta_k^{w})$$

by a successive partition of the Preisach plane $\mathscr{P}$. The parameters θ_k^{η} can be identified in the same way as in the original algorithm, i.e. by either a global least squares estimation, or by a weighted least squares estimation of only the parameters of the newly constructed partial models (see chapter 1).

Transformation of $\mathscr{P}$ Caused by the triangular shape of the Preisach plane, one could think of transforming the plane by the bijective transformation $\tau : \mathbb{R}^2 \to \mathbb{R}^2$, given by

$$(\alpha,\beta) \mapsto \tau(\alpha,\beta) := (\alpha, \beta-\alpha).$$

Then

$$\tau(\mathscr{P}) = \mathbb{R}\times\mathbb{R}_{>0}.$$

Considering a relay operator $\Gamma^{\alpha,\beta}$ for $(\alpha,\beta)\in\mathscr{P}$, the values α and β denote the lower and upper thresholds whereas the value $\beta-\alpha$ is the spread between these thresholds, and it is

intuitively clear that the relay $\Gamma^{\alpha,\beta}$ is equally well described by its lower threshold α and the spread $\beta - \alpha$. But in view of the axis-parallel partitions of the LOLIMOT algorithm, the untransformed Preisach plane $\mathscr{P}$ seems to be preferable, because the traces of the last term $F(r_{m-1}, r_m)$ in the summation formula are also axis-parallel if the input $u(t)$ varies between r_{m-2} and r_{m-1}. Indeed, in this case it is $u(t) = r_m$, and we have either $F(r_{m-1}, u(t))$ with $(r_{m-1}, u(t)) \in \mathscr{P}$ if $r_{m-1} < u(t)$ or $F(r_{m-1}, u(t)) = -F(u(t), r_{m-1})$ with $(u(t), r_{m-1}) \in \mathscr{P}$ if $u(t) < r_{m-1}$. Both cases show variation only in one variable, in the second for increasing input $u(t)$, and in the first for decreasing $u(t)$. This is not the case in the transformed Preisach plane.

Example: Identification of the hyteresis of a shock absorber As an example, figure 2.14 shows the identification of a hysteresis which is measured from a real shock absorber. The identification was done with a version of the modified LOLIMOT algorithm based on a summation formula using the primitive shapes $Q(\alpha_1, \beta, \alpha_2)$. The data have been made via so-called quasi-static measurements, i.e. the input signals where slow enough such that the dynamical effects are neglectable. More about the data in chapter 5. Problems with this identification are that the estimation is not very accurate in the details (especially on the right end), while it shows already some instability in the estimated function on the Preisach half plane (which one can already recognize from the oscillating behaviour of the estimated curve in the second picture from the left).

2.3 Conclusions

We considered two completely different types of dynamical systems:

Differential Dynamical Systems	**Hysteresis Systems**
Depend on “velocity” of input signal	Invariant under all time transformations
Local memory	Nonlocal memory
Are linear or linearization is possible	Strongly nonlinear (only exception: Hilbert transform)

At least two questions remain open:

- How to include other types of hysteresis (e.g. higher dimensional ones)?

  The Preisach model presented here accepts only one-dimensional input and output. Models for more-dimensional hysteresis are a general problem. There exist various approaches like vector Preisach, but all of them do not seem to fit well to the phenomena observed in reality.

- How to build grey-box models which are able to model coupled systems (differential and hysteresis)?

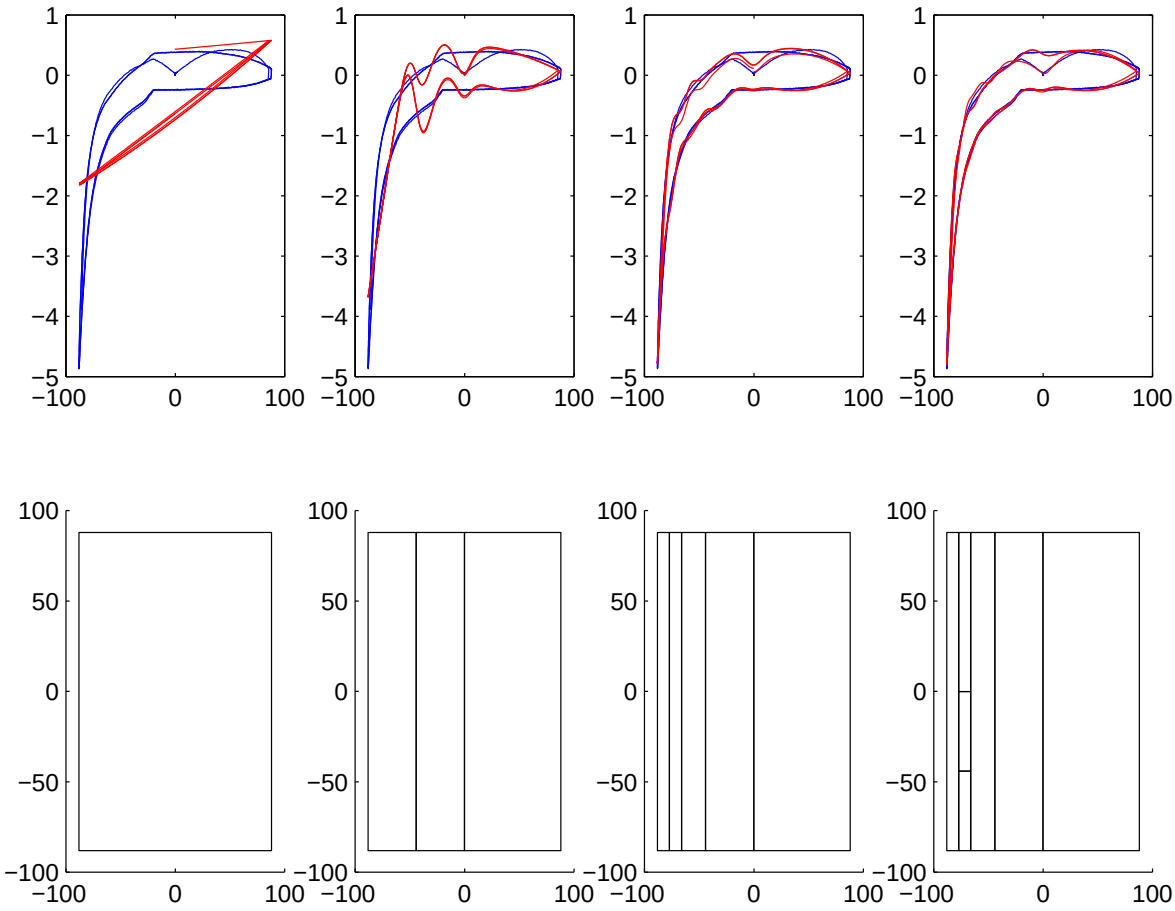


Figure 2.14: Identification of the hysteresis of a real shock absorber with second order polynomials as "local models". The upper row shows the u-y-diagrams of the measurements (blue) made on the real shock absorber together with the model output (red), with increasing number of local models ($N = 1, 3, 5, 7$ resp.) from the left to the right. The lower row shows the corresponding subdivisions of the Preisach plane.

Until now we have provided identification algorithms for both model types separately. In reality, both phenomena, i.e. nonlinear differential dynamics and hysteresis, appear mixed in some way, let's say in some "regime" we have more of the one behaviour and in some other "regime" the behaviour of the second model type is more dominant, but they still interact. An example is again the shock absorber: If it is excited with high frequencies, damping effects (viscous damping in dependence of the velocity of the excitation) are dominant, and with low frequencies one recognizes more of the hysteretic behaviour. In our example of the identification of a Preisach model for the shock absorber we have used very slow signals for the excitation. For fast signals (high frequencies), the identified model does not fit. We therefore need a model type and an identification method which are able to deal with both phenomena appearing in combination.

As a solution for both problems, we could use a summation formula which combines the summation formula of the Preisach model and the "summation formula" of the ARX models:

$$y(t) := \sum_{i=1}^{M} \sum_{j=0}^{m^{(i)}(t)-1} F(r_j^{(i)}(t), r_{j+1}^{(i)}(t)) + \sum_{k=0}^{n_u} a_k u(t-k) + \sum_{k=1}^{n_y} b_k y(t-k),$$

where the alternating sequences $(r_j^{(i)}(t))$, $i = 1,\ldots,M$ may be computed with any of the regressor components $u(t-k)$ and $y(t-k)$. Taking linearly parameterized local models for both the hysteresis and the ARX part, the parameters remain linear in the global model, and the LOLIMOT algorithm can be used without any further modifications. Of course it is not easy to decide which regressor components should be taken into the hysteresis component. And of course, the old problems remain: The inclusion of hidden states is not possible, there is no proper error model, and the LOLIMOT algorithm is just a heuristic. We did not further investigate this topic.

3 Stochastic decision theory: Bridge between theory and reality

In the previous chapter we considered deterministic models. They interpret reality as if it were free of noise, disturbances and other uncertainties. This is obviously not the case. Thus, a correct model will include also a model for these uncertainties.

There have been several attempts to model uncertainties: Probability theory, Fuzzy Logic, Dempster-Shafer theory etc. But in spite of the abundance of proposed models, it seems that the earliest of these, i.e. probability theory, is the only consistent one. This at least is the statement of Cox's theorem. To be able to use probability theory as a model for uncertainty, the interpretation of the axioms of probability has to be different from the usual "frequentist" interpretation which is based on the notion of random experiments. Instead, the appropriate interpretation is the Bayesian interpretation of probability. At first sight, it seems that it is conditional probabilities and the calculus provided by Bayes' theorem which stay in the centre of Bayesian probability. But this concerns only the formal part. At the real heart, there is the ability (and the necessity) to use these conditional probabilities, as prior probabilities or shortly priors, for actually any quantity: prior distributions are assigned to e.g. parameters and measurements, expressing the prior knowledge, belief, certainty or possibility concerning these quantities.

Bayesian theory has long time been used to describe stochastic dynamical systems, especially (stochastic) state space systems, a special case being Markov chains. Stochastic state space systems develop through hidden states which can only be observed indirectly via an additional stochastic process. A main task to do in stochastic state space models is inference on the states, called filtering. The usage of the term Bayesian probability in connection with the description of stochastic state space systems and the filtering problem is not really necessary if Bayesian probability means the *interpretation* we mentioned above: In spite of the excessive use of Bayes' theorem to filter the unobserved states through the state space model, the states can be interpreted as actually deterministic values disturbed by noise which in turn is modelled via random variables, and the filtering distribution is also a distribution in the frequentist sense. The view changes completely if the state space model is parameterized and the (unknown) parameters are equipped with prior distributions: In the frequentist view, this is not allowed: the assumption there is that there exists a "true" parameter, and this parameter thus cannot be equipped with a distribution. In contrast, in the Bayesian view, states and parameters are conceptually the same thing: random quantities with prior distributions, and Bayes' theorem allows to combine these prior distributions with observed data, thus merging both kinds of knowledge into posterior distributions. These posterior distributions contain the complete information we can extract from the prior knowledge and the data. The posterior distributions can thus be used for identification purposes: Together with a predefined loss function, we can

decide for that parameter yielding the minimal a-posteriori loss.

The most important stochastic state space models are the Hidden Markov Models (HMMs) with finite state space, and the linear Gaussian state space models with continuous state space; in the latter systems, states and observations are propagated by linear functions and with Gaussian random noise. The filter (i.e. the estimator for the hidden states) in both cases can be computed analytically. In the case of linear Gaussian state space systems, it is the Kalman filter. In practically all other cases, i.e. if the state space model has no finite state space or is not linear or not Gaussian, the filter cannot be given analytically: high-dimensional integrals have to be computed where a closed-form solution does not exist. They have to be approximated. The high dimensionality of the integrals prohibits the use of the usual grid-based numerical methods: the difficulty (complexity) of computation increases exponentially with the dimension. An alternative are Monte Carlo methods: In one dimension, they are slower than grid-based methods, but the complexity remains the same with increasing dimensionality, at least in the optimal case. In non-dynamical settings, Markov Chain Monte Carlo (MCMC) methods have been established as a generic tool for the computation of complex distributions: to be mentioned are the Gibbs sampler and the Metropolis-Hastings algorithms. With state space systems, it is better to use recursive methods and to break down the high-dimensional integrals into numerous but lower-dimensional ones. Monte Carlo methods based on recursive procedures are the Sequential Monte Carlo (SMC) methods, sometimes called particle filters.

Overview We will first describe roughly the usage of probability theory to model uncertainty. We then provide the basic definitions and concepts of Bayesian probability theory and stochastic decision theory, followed by a short description of the efforts made for their justification. The next section describes strategies for the elicitation of prior distributions. We then proceed by considering general stochastic models. After summarizing the computational possibilities for the approximative computation of complex distributions given by Monte Carlo methods, we focus our look on stochastic state space models and the recursive Monte Carlo methods.

Contributions This chapter is mainly an overview and introduction into topics of stochastic decision theory and stochastic state space systems. It combines information from several sources, but nothing is new. The aim is the preparation and justification of methods used for our model and identification scheme in chapter 5.

3.1 Models for reality

An omnipresent task we encounter in our daily life is the necessity to predict or forecast something concerning a ***real system***, may it be the outcome of an action, the well- or malfunctioning of a machine, or the weather. The scientific way to produce predictions and forecasts is the use of a model. A model will let us gain information otherwise not accessible for some reason, be it because it is an “internal signal” of the real system which is not measurable, or be it that it is something concerning the future. Especially when treating physical, chemical, biological or even social systems, *mathematical* modelling has shown to be a powerful tool.

Modelling approach The task of treating real problems with mathematical modelling uses two separated steps (see Mumford and Desolneux [in preparation]):

- Create a (stochastic) model and verify it (***modelling***);
- Seek for an algorithm for applying models to practical problems (***computation***).

There is a third step, completing the "triad" (according to Samarskii, see Neunzert and Rosenberger [1993]):

- Make a computer program which implements the desired algorithm (***implementation***).

In this chapter, we focus first on modelling, later something will be said about computation. Implementation issues will be put aside until chapter 5.

System, model, and interpretation What is a ***mathematical model***? We want to understand it as:

- A representation of a real system in mathematical terms, focussing only on some interesting aspects of the real system.

A model can never account for all aspects of reality. Nevertheless, apart from the necessary restrictions, the mathematical logic of the model shall follow the natural logic of the real system. The model is thus an image of (a part of) the real system. For one real system $\mathscr{S}$ infinitely many models are possible. It is better to look at it the other way round: To each model $\mathscr{M}$ belongs a map which connects the model to the real system. This map is called the ***interpretation*** of the model:

$$\mathscr{J} : \mathscr{M} \longrightarrow \mathscr{S}.$$

For each model $\mathscr{M}$, infinitely many interpretations exist. Concerning the interpretation $\mathscr{J}$, one has to be aware that the model itself is mathematics, so mathematical logic reigns: statements can principally be proved to be or not to be valid. In contrast, the interpretation is the connection between model and system, thus between mathematics and reality, and stands therefore outside of mathematics: It never can be proved mathematically that some chosen interpretation is the "correct" one. Quarrels may arise about the appropriate interpretation, and this is legitime. A mathematical model together with its interpretation must always be verified against reality by experiments. There is always a "range of validity" concerning aspects of the system, knowledge on the system (which may even vary over time), etc.

Uncertainties But how can a model be verified? This has to be done by making experiments on the real system and by collecting measured data. Here, a fundamental problem arises: We never can be sure about the collected data. We are concerned with ***measurement errors*** and ***measurement noise***. So, for example, if our measurement is given by a real value, this value is determined by infinitely many digits, but we can measure only finitely many, and measurements are always disturbed by noise. Additionally, we have ***model errors***, because, as already mentioned, we are never able to construct a model which exactly describes the

given real system in all respects. We always need to simplify and to focus on those aspects of the real system which we are interested in. But there are other aspects which influence the system behaviour. There is furthermore the outer world, which also influences the outcomes of the experiments. We have ***imperfect knowledge***: we are not able to know all aspects of a real system, and we are thus not able to plug them into the model. And even if our model were absolutely correct, we still would have uncertainties about the initial or actual state of our real system, information we can only gain by measurements. We have to cope with these uncertainties, and the correct way to handle these uncertainties is by including them into our model, and our algorithms must be prepared to handle them. Probability theory provides the necessary instruments and methods.

Deterministic vs. stochastic modelling An instructive example for what happens when such disturbances are being neglected is given in Ljung [1999], section 3.3:

Example: Consider the *deterministic* model given by

$$y(t) = b\sum_{k=1}^{\infty} a^{k-1}u(t-k)$$

with parameters $a, b \in \mathbb{R}$. Using the shift operator

$$(q^{-1}u)(t) = u(t-1)$$

we can write this in operator notation as (geometric series)

$$y(t) = \frac{bq^{-1}}{1-aq^{-1}}u(t).$$

We can equally well write

$$(1-aq^{-1})y(t) = bq^{-1}u(t),$$

i.e.

$$y(t) - ay(t-1) = bu(t-1).$$

Let now be the data $y(s)$, $u(s)$ for $s \leq t-1$ be given. If both data and system description are correct, the "predictors" for $y(t)$ given by either

$$\hat{y}(t\,|\,t-1) := b\sum_{k=1}^{\infty} a^{k-1}u(t-k)$$

or

$$\hat{y}(t\,|\,t-1) := ay(t-1) + bu(t-1)$$

are completely equal. But with inclomplete or disturbed data, they are vulnerable to different imperfections: If input-output data are lacking prior to time $s = 0$, then the first predictor suffers from an error that decays like a^t (wrong initial conditions), whereas the second predictor is still correct for time $t \geq 1$. On the other hand, if the output data are disturbed by measurement errors, the first predictor is unaffected, whereas such errors are directly transferred into the prediction of the second predictor. If the model had been complemented with a proper noise model (and a loss function), then the choice of the predictor would have become unique.

The deterministic model leads to several equivalent descriptions yielding different algorithms. But some are stable and some are unstable. Modelling with stochastic models automatically leads to the right algorithm. Deterministic model actually means: the measured data are exact, there is no noise; this is never true in reality.

Moving from deterministic to stochastic systems The way to deal with the unavoidable uncertainties of reality is to use stochastic (probabilistic) models, and to put distributions on the state and the outputs of our dynamical system. Until now we have only considered deterministic systems: same input and same initial conditions lead always to the same output. This is guaranteed by the Causality Axiom and the Cocylce Property of chapter 2. Since we need to introduce uncertainty into our models, we thus have to introduce a certain random behaviour into the dynamical system, and cannot maintain these two axioms. We have to replace them by some conditions on the distributions of the states and the outputs given some input and initial conditions in such a way that, if we consider the special case of determinism, then the system reduces to a deterministic dynamical system in the sense of chapter 2. Determinism means here, that the occurring distributions are all Dirac distributions. One way to handle random systems are random processes. Causality Axiom and Cocycle Property and even the Consistency axiom can easily seen generalized when looking at the definition of a random process. But there are two important differences: With random processes the focus lies on the randomness, not on the input-output behaviour like in Systems and Control Theory. Therefore inputs are only introduced in a second step, as covariates. The other difference concerns the Interval axiom. Usually processes are not allowed to explode or to die out. The Interval Axiom was introduced exactly to handle systems which show this behaviour. Closely related is stability, and in connection with Markov chains, we will mention some stability concepts in subsection 3.4.3.

Stochastical modelling We put distributions on states and outputs of our systems. But our task is mainly identification of unknown parameters, and uncertainty on states and outputs infers also uncertainty on identified parameters. If we do not know real parameters for sure, we could at least try to say something about the probability of their outcomes: We deal with distributions of parameters. These distributions are of course as well subject to uncertainties, but the uncertainties are in some respect pushed a step farther away (hierarchical models do this repeatedly). We model our uncertainty by stochastical means; or even more rigourous: we use stochastics as a model for uncertainty. This is precisely the interpretation of stochastics used in Bayesian probability theory. As said before, to each model an interpretation belongs: "uncertainty" is the real system, "probability theory" is the model, and "Bayesianism" is the interpretation, the map between model and reality. There have been and are still quarrels about this interpretation, precisely about the Bayesian interpretation of probability theory as a model for uncertainty.

Frequentist and Bayesian interpretation of probability Opposed to this is the "usual" way to interpret probability theory represented by the "frequentist" interpretation:

- Probabilities are only defined for "random experiments", *not* for (individual) parameters.

The Bayesian interpretation is much more comprehensive and general, and in this sense much more useful: it serves as a model for much more parts of reality.

With Bayesian probability theory, probabilities are always conditioned: $\Pr(E\,|\,C)$ is a measure of the (presumable rational) belief in the occurrence of the event E under conditions C (Bernardo [2003]). Unconditioned probabilities do not exist. The conditions C are in most applications given by:

- ***Assumptions*** A: Model Assumptions,
- ***Knowledge*** K: System Knowledge,
- ***Data*** D: Measured Data.

One usually drops the conditioning on the assumptions A and the knowledge K in the notations, and writes down only the data D explicitly.

Subjectivity versus objectivity One of the main arguments against Bayesian probability is the introduction of subjectivity into the statistical inference. Notions like "belief", "prior knowledge", "uncertainty" are considered to be subjective: each person has its own belief, knowledge etc. Thus, Bayesian Probability is prone to subjectivity, as the argument goes. Frequentist probability would be objective. But Bayesian Probability makes all assumptions explicit. Frequentist probability is also based on assumptions, but they are not made explicit, and are often not justified for the practical problem at hand. In Bayesian Statistics, there are furthermore attempts to construct "objective" priors, so-called non-informative priors.

Decision making The purpose of modelling is in most cases to make decisions in the real world. These decisions should be done in two steps: First, given prior knowledge, assumptions and data, we infer posterior knowledge. Therefore some kind of logic is needed. Second, use the inferred knowledge to choose an action. Thus, one has to decide for a best action. In the Bayesian approach, the prior knowledge is given by distributions of parameters, the assumptions consist of the choice of the stochastic model, and the data are assumed to be noisy and modelled by distributions; the inference is done by (the Generalized) Bayes' Rule (Jaynes [1990]), and the choice of the action is done by optimizing a loss or utility function.

Bayesian probability theory as logic Thus, some kind of logic is needed. ***Logic*** is a model for "real world reasoning". The interpretation map has to point on notions like "follows", "and", "not" etc. As with all models of reality, one has to accept an appropriate logic, depending on the given problem. As always, there does not exist *the* right logic for all possible purposes. The special problem here is the reasoning under uncertainty. Thus, Aristotelian logic is not enough, because there all statements are either true or false, and nothing else. So, one seeks for other kinds of logic. Many candidates have been proposed: Bayes probability, Fuzzy Logic, Dempster-Shafer (DS) logic, etc. But in spite of this large variety of

proposed logics, there are reasons to believe that Bayes probability is the most justified model for reasoning under uncertainty.

Decision as action At the end, an action has to be done, a decision has to be made. Concerning conditional probabilities, we want to note that an important distinction should be made between two types of conditioning (see e.g. Lauritzen [2001]): An *action done* given some conditions is different from an *event observed* under certain conditions. Lauritzen [2001] refers to these respectively as ***conditioning by intervention***, with the notation

$$p(x\|y) = \Pr(X = x | Y \leftarrow y),$$

and ***conditioning by observation***, written

$$p(x|y) = \Pr(X = x | Y = y).$$

Many misunderstandings result from an unallowed confusion of these two kinds of conditional probabilities. (These misunderstandings are then used in turn as arguments against Bayesian probability.)

We will consider the estimation of the parameters of a given parameterized model as the making of a decision. The theory which considers decision making is ***(stochastic) decision theory***. We will go into more detail on Bayesian probability theory and decision theory in the next section.

3.2 Bayesian statistics

Let in the following X and Y be random variables for a suitable measure space over arbitrary sets $\mathscr{X}$ and $\mathscr{Y}$ respectively, and let $x \in \mathscr{X}$ and $y \in \mathscr{Y}$ be realizations of X and Y. We will usually drop the random variables from our notations respectively do not distinguish between the random variables and their realizations (this is in accordance with e.g. Robert [2001]; also Ljung [1999] uses explicitly the same notation for stochastic processes and their realizations, see there section 2.1). This avoids also the problem that occurs if the parameter θ, viewed as a random variable (in the Bayesian way), needs a symbol for this random variable, whereas the greek upper case letter Θ is usually reserved for the parameter set. We also use the following common notations:

- $x|y \sim p(x|y)$ if x given y is distributed according to $p(x|y)$, and
- $g(x) \propto h(x)$ if $g(x)$ is proportional to $h(x)$, i.e. if there is a constant $C \in \mathbb{R}_{>0}$ such that

$$g(x) = Ch(x).$$

We often implicitly assume a reference measure μ for the occurring densities to be given; in the discrete case this is actually always the counting measure, in the continuous real case it is usually the Lebesgue measure. We have thus

$$\Pr(x \in A) = \int_A p(x) d\mu(x) = \begin{cases} \int_{x \in A} p(x) dx & \text{(continuous case),} \\ \sum_{x \in A} p(x) & \text{(discrete case),} \end{cases}$$

if $x \sim p(x)$. We also write $\mathbf{E}^p[h(x)]$ for the expectation of $h(x)$ if x is distributed according to p (omitting p in the notation if it is clear from the context):

$$\mathbf{E}^p[h(x)] := \int_{\mathscr{X}} h(x)p(x)d\mu(x).$$

We adopt also the short notation $y_{1:n}$ for the ensemble of values y_i, $i = 1,\ldots,n$, often used with distributions or densities like

$$p(x|y_{1:n}) = p(x|y_1,\ldots,y_n).$$

3.2.1 Bayes' theorem

We follow closely Robert [2001].

Parametric models In the following we assume $\mathscr{Y}$ to be a set and that ***observations*** $(y_1,\ldots,y_n)$ with $y_i \in \mathscr{Y}$ are given which are generated according to a probability distribution

$$f_i(y_i\,|\,\theta_i,y_1,\ldots,y_{i-1}) = f_i(y_i\,|\,\theta_i,y_{1:i-1}),$$

where $\theta_i \in \Theta_i$ are some parameters coming from parameter sets Θ_i. Then the ***sample density*** is given as the joint distribution of all observations $y = (y_1,\ldots,y_n)$ given $\theta = (\theta_1,\ldots,\theta_n)$

$$f(y\,|\,\theta) = \prod_{i=1}^{n} f_i(y_i\,|\,\theta_i,y_{1:i-1})$$

with $\theta := (\theta_1,\ldots,\theta_n) \in \Theta := \Theta_1 \times \cdots \times \Theta_n$. In the examples for this section, we usually restrict investigations to the case where only one observation $y \in \mathbb{R}$ is given. The case with several observations $y_1,\ldots,y_n \in \mathbb{R}$ can usually be reduced to the previous situation through a sufficient statistic. The following defintition can be found e.g. in Robert [2001]:

Definition 3.1 (Parametric Statistical Model)**:** *A **parametric statistical model** consists of the observation of a random variable y, distributed according to $f(y\,|\,\theta)$, where only the parameter θ is unknown and belongs to a vector space Θ of finite dimension.*

If the sampling distribution is viewed as a function on θ, i.e.

$$\ell(\theta\,|y) = f(y\,|\,\theta),$$

it is called the associated ***likelihood*** (nevertheless, we usually do not distinguish between them and use the terms "model", "sample distribution" and "likelihood" synonymously). If, for given observations $y_1,\ldots,y_n$, one wants to infer some knowledge about the unknown parameter θ, one has to invert probabilities. This is accomplished by Bayes' theorem:

Bayes' theorem Given two events A and E, the definition of conditional probability relates $\Pr(A|E)$ and $\Pr(E|A)$ by

$$\Pr(A|E)\Pr(E) = \Pr(A,E) = \Pr(E|A)\Pr(A).$$

If $\Pr(E) \neq 0$, we can divide by $\Pr(E)$, yielding ***Bayes' theorem***:

$$\Pr(A|E) = \frac{\Pr(E|A)\Pr(A)}{\Pr(E)}.$$

Marginalization Often, $\Pr(E)$ has to be computed by ***marginalization***, i.e. by eliminating a variable of a joint distribution via integration:

$$\Pr(E) = \int \Pr(A,E)dA = \int \Pr(E|A)\Pr(A)dA.$$

The Bayesian way to model the uncertainty of the parameters $\theta \in \Theta$ is by means of a probability distribution $\pi(\theta)$ on Θ, called ***prior distribution***. All inference is based on the distribution of θ conditional on y, $\pi(\theta|y)$, called the ***posterior distribution***. Bayes' theorem yields for this posterior distribution:

$$\pi(\theta|y) := \frac{f(y|\theta)\pi(\theta)}{\int f(y|\theta)\pi(\theta)d\theta}.$$

Altogether we get (see Robert [2001]):

Definition 3.2: *A **Bayesian statistical model** is made of a parametric statistical model*

$$(\mathscr{Y}, f(y|\theta))$$

and a prior distribution on the parameters,

$$(\Theta, \pi(\theta)).$$

Given a parametric model $f(y|\theta)$ and a prior distribution $\pi(\theta)$, several distributions are of interest (see Robert [2001]):

- the ***joint distribution*** of (y,θ):

$$\varphi(y,\theta) := f(y|\theta)\pi(\theta),$$

- the ***marginal distribution*** of y:

$$m(y) := \int_\Theta \varphi(y,\theta)d\theta = \int_\Theta f(y|\theta)\pi(\theta)d\theta,$$

- the ***posterior distribution*** of θ:

$$\pi(\theta|y) := \frac{f(y|\theta)\pi(\theta)}{\int_\Theta f(y|\theta)\pi(\theta)d\theta} = \frac{f(y|\theta)\pi(\theta)}{m(y)},$$

- the ***predictive distribution*** of z when $z \sim g(z \mid \theta, y)$:

$$g(z \mid y) := \int_\Theta g(z \mid \theta, y)\pi(\theta \mid y)d\theta.$$

In the following, we write

- $\mathbf{E}_\theta[h(y)]$ for the expectation of $h(y)$ under the distribution $y \sim f(y \mid \theta)$, and
- $\mathbf{E}^\pi[h(\theta) \mid y]$ for the expectation of $h(\theta)$ under the posterior distribution of θ, $\pi(\theta \mid y)$, given the prior π.

Especially in so-called non-informative settings (see section 3.3.1), where the prior is considered to represent as few information as possible, it is often necessary to allow not only a probability distribution as prior, i.e. measures π such that

$$\int_\Theta \pi(\theta)d\theta = 1,$$

in which case the prior is called ***proper***, but to extend the possible priors to σ-finite measures π such that

$$\int_\Theta \pi(\theta)d\theta = +\infty.$$

Then π is called an ***improper prior distribution***. The extension of the posterior distribution $\pi(\theta \mid y)$ associated with an improper prior π is then given by the ***Generalized Bayes Formula***:

$$\pi(\theta \mid y) = \frac{f(y \mid \theta)\pi(\theta)}{\int_\Theta f(y \mid \theta)\pi(\theta)d\theta} \qquad \text{as far as} \qquad \int_\Theta f(y \mid \theta)\pi(\theta)d\theta < \infty.$$

One can justify improper priors by considering them as limits of proper priors.

3.2.2 Foundations of decision theory

Statistics is used to make real world decisions. The Bayesian approach is much more suited than the frequentist approach if we want to include prior knowledge explicitly. We always have to include prior knowledge, but in the frequentist case it is often done implicitly (compare for example Jaynes [1976]). In the Bayesian context, prior knowledge is included by prior distributions.

The decisions most often taken by statisticians are estimations and tests, e.g.

- point estimation,
- interval estimation,
- hypothesis tests.

If one estimates a parameter, one could choose as estimator every value from a given parameter set. Which of those possible estimators is preferred depends on the action one is intended to do. The theory which investigates this is called Decision Theory. To formalize precisely what a "good" choice is, one has to define a loss function which is to be minimized. Known loss functions in frequentist settings are the quadratic loss (for point estimation) and the 0–1 loss (for hypothesis tests). A loss function which is invariant against parameter transformations is often to be preferred. One loss function with this property is the intrinsic loss. In the following, we want to exploit the Bayesian Decision Theory and its relation to frequentist notions in more detail. We follow Robert [2001] and Berger [1980].

Decision rules, loss functions and estimators

Decisions In Decision Theory, our aim is to choose a ***decision*** among a set of possible decisions. Generally, decisions are called ***actions***. We therefore (following e.g. Berger [1980]) denote them by $a \in \mathscr{A}$ where $\mathscr{A}$ shall denote the set of possible actions. For ***estimation*** purposes, i.e. if an unknown parameter θ (also called "***state of nature***") is searched for, the set $\mathscr{A}$ is often chosen to be equal to the parameter set Θ, i.e. $\mathscr{A} = \Theta$. In general, we need a ***decision procedure*** or ***decision rule*** $\delta : \mathscr{Y} \longrightarrow \mathscr{A}$ assigning to each observation $y \in \mathscr{Y}$ a corresponding decision (action) $\delta(y) \in \mathscr{A}$. In estimation problems, i.e. if $\mathscr{A} = \Theta$, the decision procedure δ will be called ***estimator*** and the value $\delta(y) \in \Theta$ will be called ***esimate*** (of the parameter θ).

Randomized decisions Sometimes, for practical as well as theoretical reasons, one considers so-called randomized decision rules (see e.g. Berger [1980]). A ***randomized decision rule*** $\delta^*(y,\cdot)$ is, for each $y \in \mathscr{Y}$, a probability distribution on $\mathscr{A}$. The interpretation is, that if y is observed, $\delta^*(x,A)$ is the probability that an action in $A \subseteq \mathscr{A}$ will be chosen. In contrast, decision rules $\delta : \mathscr{Y} \longrightarrow \mathscr{A}$ are called ***non-randomized***. These can be seen as special cases of randomized decision rules (applying a delta distribution). Nevertheless, we want to consider exclusively non-randomized decision rules, but the theory applies as well to randomized decision rules mostly without changes.

Loss functions In the case of a parameterized model $f(y|\theta)$, the decision rule is build via an evaluation criterion for parameters $\theta \in \Theta$ and decisions $a \in \mathscr{A}$. This criterion is modelled by a ***loss function*** $L(\theta, a)$:

$$L : \Theta \times \mathscr{A} \longrightarrow \mathbb{R}_{\geq 0}$$

which models the loss one gets if the unknown parameter is equal to θ and the action/decision a is chosen.

Decision procedures Especially ***Bayesian Decision Theory*** deals thus with the following three maps (see Robert [2001]):

(1) On $\mathscr{Y}$: Distribution for the observation, $f(y|\theta)$,

(2) On Θ: Prior distribution for the parameter, $\pi(\theta)$,

(3) On $\Theta \times \mathscr{A}$: Loss function associated with the decisions, $L(\theta, a)$.

These three maps are the basis for the determination of a decision procedure. We want to make our decision $a \in \mathscr{A}$ in such a way that the loss function $L(\theta, a)$ is minimal for a given θ. Therefore only loss functions with

$$L(\theta, a) \geq K > -\infty$$

for some $K \in \mathbb{R}$ will be considered. If the parameter θ is unknown, then it is generally impossible to minimize the loss function uniformly with respect to θ and a. Frequentist and Bayesian probability follow different principles to determine a decision procedure; a major rôle in both cases is played by the average loss or frequentist risk

$$R(\theta, \delta) = \mathbf{E}_\theta[L(\theta, \delta(y))] = \int_{\mathscr{Y}} L(\theta, \delta(y)) f(y \,|\, \theta) dy.$$

In the following, we consider only loss functions δ where this risk is finite:

$$\mathscr{D} := \left\{ \delta \;\middle|\; R(\theta, \delta) < \infty \right\}.$$

We follow Robert [2001]:

- The ***frequentist principle*** is based on the ***average loss*** or ***frequentist risk***

  $$R(\theta, \delta) = \mathbf{E}_\theta[L(\theta, \delta(y))] = \int_{\mathscr{Y}} L(\theta, \delta(y)) f(y \,|\, \theta) dy.$$

  With the frequentist risk, the error is averaged over all values of y proportionally to $f(y \,|\, \theta)$. The problem, which is often encountered in practice, is that this is not the best choice for some individual data y. Additionally, frequentist probability bases on the assumption of the repeatability of the experiment, which is not always justified. Another problem is that this principle does not induce a total preorder on the set of decision rules, i.e. not all decision rules are comparable: there may be decision rules δ_1 and δ_2 as well as parameters θ_1 and θ_2 such that

  $$R(\theta_1, \delta_1) < R(\theta_1, \delta_2) \qquad \text{but} \qquad R(\theta_2, \delta_1) > R(\theta_2, \delta_2)$$

  ("crossing").

- The ***Bayesian principle*** is to integrate over the space Θ of parameters to get the ***posterior expected loss***

  $$\rho(\pi, a \,|\, y) = \mathbf{E}^\pi[L(\theta, a) \,|\, y] = \int_\Theta L(\theta, a) \pi(\theta \,|\, y) d\theta.$$

  An alternative way to proceed is to integrate over the space Θ while weighting the risk R by the prior π and to compute thus the ***Bayes risk***

  $$r(\pi, \delta) = \mathbf{E}^\pi[R(\theta, \delta)] = \int_\Theta \int_{\mathscr{Y}} L(\theta, \delta(y)) f(y \,|\, \theta) dy \pi(\theta) d\theta$$

which induces a total preordering on the decision rules. This ensures that two decision rules are always comparable. An estimator minimizing $r(\pi,\delta)$ can be obtained by selecting, for every $y \in \mathscr{Y}$, the value $\delta(y)$ which minimizes $\rho(\pi, a\,|\,y)$ since

$$r(\pi,\delta) = \int_{\mathscr{Y}} \rho\big(\pi,\delta(y)\,|\,y\big)m(y)dy,$$

if it exists.

Thus, both approaches to the Bayesian principle give the same decision rule (see e.g. Robert [2001]):

Definition 3.3: *If a decsision rule δ^π exists which minimizes $r(\pi,\delta)$,*

$$\delta^\pi := \arg\min_\delta r(\pi,\delta),$$

*then each such δ^π is called a **Bayes rule** (associated with a prior distribution π and a loss function L). The value $r(\pi) := r(\pi,\delta^\pi)$ is called the **Bayes risk**.*

This definition is valid for both proper and improper priors in all cases where $r(\pi) < \infty$. Otherwise, we define the ***Generalized Bayes Rule*** pointwise:

$$\delta^\pi := \arg\min_{a\in\mathscr{A}} \rho(\pi, a\,|\,y)$$

if $\rho(\pi,a\,|\,y)$ is well-defined for every y. (One should not confuse "Generalized Bayes" and "Improper Bayes").

Minimaxity and admissibility

We want now to describe formal relations between frequentist and Bayes principles and to shortly relate the Bayesian notions defined above to well-known frequentist notions, precisely minimaxity and admissibility (see again Robert [2001] and Berger [1980]).

Minimaxity

Definition 3.4: *The **minimax risk** associated with a loss L is*

$$\overline{R} = \inf_{\delta\in\mathscr{D}} \sup_\theta R(\theta,\delta) = \inf_{\delta\in\mathscr{D}} \sup_\theta \mathbf{E}_\theta[L(\theta,\delta(y))],$$

*and a **minimax rule** is any rule δ^{M} such that*

$$\sup_\theta R(\theta,\delta^{\mathrm{M}}) = \overline{R}.$$

The minimax risk introduces a total preordering on $\mathscr{D}$ and insures against the worst case. This worst case reasoning leads often to very conservative estimators and a-priori knowledge cannot be included to reveal less conservative estimators. The existence of the minimax estimator can be ensured in quite many cases (see e.g. Robert [2001]):

Theorem 3.1: *If $\mathscr{A} \subseteq \mathbb{R}^k$ is convex and compact, and if $L(\theta, a)$ is continuous and convex as a function of a for every $\theta \in \Theta$, then there exists a (non-randomized) minimax estimator.*

Now, the Bayes risks are never greater than the minimax risk:

$$\underline{r} := \sup_{\pi} r(\pi) = \sup_{\pi} \inf_{\delta \in \mathscr{D}} r(\pi, \delta) \leq \overline{R} = \inf_{\delta \in \mathscr{D}} \sup_{\theta} R(\theta, \delta)$$

where we call $\underline{r}$ the ***maximin risk***: it is associated with the least favourable prior. One defines (Robert [2001]):

Definition 3.5: *The decision problem **has a value** if $\underline{r} = \overline{r}$, i.e.*

$$\sup_{\pi} \inf_{\delta \in \mathscr{D}} r(\pi, \delta) = \inf_{\delta \in \mathscr{D}} \sup_{\theta} R(\theta, \delta).$$

We see: If the decision problem has a value, then some minimax estimators are Bayes estimators for the least favourable distributions.

To check for minimaxity in connection with Bayes rules, the following holds (see e.g. Robert [2001]):

Theorem 3.2: ***(i)*** *If δ_0 is a Bayes rule for the prior π_0 and if $R(\theta, \delta_0) \leq r(\pi_0)$ for every θ in the support of π_0, then δ_0 is minimax and π_0 is the least favourable distribution.*

(ii) *If for a sequence (π_n) of proper priors the generalized Bayes estimator δ_0 satisfies*

$$R(\theta, \delta_0) \leq \lim_{n \to \infty} r(\pi_n) < +\infty$$

for every $\theta \in \Theta$, then δ_0 is minimax.

Admissibility The second important frequentist decision principle is given by the admissibility of decision rules:

Definition 3.6: *A decision rule δ_0 is called **inadmissible** if there exists a decision rule δ_1, such that for every θ,*

$$R(\theta, \delta_0) \geq R(\theta, \delta_1)$$

and, for at least one θ_0,

$$R(\theta_0, \delta_0) > R(\theta_0, \delta_1).$$

*Else, the decision rule is called **admissible**.*

Admissibility reduces the set of decision rules based on *local* properties (as opposed to the minimax rules). Relations between minimaxity and admissibility are (Robert [2001]):

Theorem 3.3: ***(i)*** *If there exists a unique minimax estimator, then this estimator is admissible. The converse is false.*

(ii) *If δ_0 is admissible with constant risk, then δ_0 is the unique minimax estimator. The converse is false.*

Admissibility is strongly related to the Bayesian paradigm (Robert [2001]):

Theorem 3.4: ***(i)*** *If π is strictly positive on Θ, with*

$$r(\pi) = \int_\Theta R(\theta, \delta^\pi)\pi(\theta)d\theta < \infty$$

and $R(\theta, \delta)$ is continuous, then the Bayes rule δ^π is admissible.

(ii) *If the Bayes rule associated with the prior π is unique, then it is admissible.*

Thus, Bayes rules are virtually always admissible. The reason is that, if a rule with better frequentist risk $R(\theta, \delta)$ existed, the rule would also have better Bayes risk $\mathbf{E}^\pi[R(\theta, \delta)]$. Actually, even more can be said: Bayes estimators often constitute the class of admissible estimators. In contrast, Bayes estimators may be inadmissible when the Bayes risk is infinite. But for a bounded loss, the Bayes risk is clearly finite.

More complicated is the case of generalized Bayes estimators. One situation in which the Generalized Bayes rule δ can be shown to be admissible is when the loss is positive and the Bayes risk $r(\pi, \delta)$ finite. Unfortunately, it is rather rare to have finite Bayes risk in the case of improper π. This makes the verification of admissibility or inadmissibility very difficult (see e.g. Berger [1980], section 4.5).

Usual loss functions and their Bayes rules

We shortly summarize the most important loss functions. Most of them are well-known in the frequentist sense; they can also be applied to the Bayesian principle. We follow again Robert [2001] and Berger [1980].

- ***Quadratic loss*** for $\mathcal{A} = \Theta = \mathbb{R}$:

  $$L(\theta, a) := (\theta - a)^2.$$

  The quadratic loss is the most common loss function. It was proposed by Legendre (1805) and Gauss (1810). The Bayes rule (Bayes estimator) associated with a prior π and the quadratic loss is the posterior expectation (posterior mean)

  $$\delta^\pi(y) = \mathbf{E}^\pi[\theta\,|y] = \frac{\int_\Theta \theta\, f(y\,|\,\theta)\pi(\theta)d\theta}{\int_\Theta f(y\,|\,\theta)\pi(\theta)d\theta}.$$

  For the multi-dimensional case $\mathcal{A} = \Theta = \mathbb{R}^d$, the quadratic loss is given by

  $$L(\theta, a) := (\theta - a)^\top Q\,(\theta - a)$$

  whith a positive definite symmetric $d \times d$ matrix Q. The Bayes estimator is in this case also the posterior mean $\delta^\pi(y) = \mathbf{E}^\pi[\theta\,|y]$.

- ***Absolute error loss*** for $\mathscr{A} = \Theta$ (Laplace, 1773):

$$L(\theta, a) := |\theta - a| \qquad \text{or} \qquad L_{k_1,k_2}(\theta, a) := \begin{cases} k_2(\theta - a) & \text{if } \theta > a, \\ k_1(a - \theta) & \text{otherwise.} \end{cases}$$

  The Bayes estimator associated with a prior π and L_{k_1,k_2} is a quantile of order $(k_2/(k_1 + k_2))$ of $\pi(\theta\,|\,y)$. For $k_1 = k_2$, we get the absolut error loss, and the Bayes estimator is the median of $\pi(\theta\,|\,y)$, i.e. the $1/2$-quantile of the posterior.

- ***0-1 loss*** (Neyman-Pearson loss for testing hypotheses): We want to test the hypothesis H_0: $\theta \in \Theta_0$ against the hypothesis H_1: $\theta \notin \Theta_0$. Thus, the decision set is chosen to be $\mathscr{A} := \{0, 1\}$ where $a = 1$ if H_0 is accepted. The loss function is defined as

$$L_{0\text{-}1}(\theta, a) := \begin{cases} 1 - a & \text{if } \theta \in \Theta_0, \\ a & \text{otherwise,} \end{cases}$$

  associated with the risk (type-one and type-two errors)

$$R(\theta, \delta) = \mathbf{E}_\theta[L(\theta, \delta(y))] = \begin{cases} \Pr_\theta(\delta(y) = 0) & \text{if } \theta \in \Theta_0, \\ \Pr_\theta(\delta(y) = 1) & \text{otherwise.} \end{cases}$$

  The Bayes rule associated with π and 0-1 loss is

$$\delta^\pi(y) = \begin{cases} 1 & \text{if } \Pr(\theta \in \Theta_0\,|\,y) > \Pr(\theta \notin \Theta_0\,|\,y), \\ 0 & \text{otherwise.} \end{cases}$$

- ***Intrinsic losses*** again for the case $\mathscr{A} = \Theta$: The choice of the parameterization is important because, contrary to the maximum likelihood estimation approach, if ϕ is a one-to-one transformation of θ, the Bayes estimator of $\phi(\theta)$ is usually different from the transformation by ϕ of the Bayes estimator of θ under the same loss. This is a problem in noninformative settings without natural parameterization. In this case, one wishes that the estimators should be invariant under reparameterization ("ultimate invariance"). The corresponding parameterization-free loss functions compare directly the distributions $f(\cdot\,|\,\theta)$ and $f(\cdot\,|\,a)$ using some distribution distance d:

$$L(\theta, a) = d\big(f(\cdot\,|\,\theta), f(\cdot\,|\,a)\big).$$

  There are two usual distribution distances:

  (1) ***Entropy distance*** or ***Kullback-Leibler divergence***:

$$L_{\mathrm{e}}(\theta, a) = \mathbf{E}_\theta\left[\log\left(\frac{f(y\,|\,\theta)}{f(y\,|\,a)}\right)\right] = \int_{\mathscr{Y}} f(y\,|\,\theta)\log\left(\frac{f(y\,|\,\theta)}{f(y\,|\,a)}\right) dy.$$

  (This is not a distance in the mathematical sense because of its asymmetry.)

  (2) ***(Squared) Hellinger distance***:

$$L_{\mathrm{H}}(\theta, a) = \frac{1}{2}\mathbf{E}_\theta\left[\left(\sqrt{\frac{f(y\,|\,a)}{f(y\,|\,\theta)}} - 1\right)^2\right] = \frac{1}{2}\int_{\mathscr{Y}}\left(\sqrt{f(y\,|\,a)} - \sqrt{f(y\,|\,\theta)}\right)^2 dy.$$

Considering the normal case where $\pi(\theta\,|\,y)$ is a $\mathcal{N}(\mu(y),\sigma^2)$ distribution, the Bayes estimator is $\delta^\pi(y)=\mu(y)$ in both cases. Whereas the Hellinger loss may be preferable beacuse it always exists, it does not lead to explicit Bayes estimators except in the normal case. On the contrary, in exponential families (see subsection 3.3.1), the entropy loss provides explicit estimators which are the posterior expectations for the estimation of the natural parameter.

Bayesian point estimation without loss function

For point estimation with no loss function, one considers the posterior distribution

$$\pi(\theta\,|\,y)\propto f(y\,|\,\theta)\pi(\theta)=\ell(\theta\,|\,y)\pi(\theta).$$

This gives the summary of the information available on θ by integrating simultaneously prior information *and* information brought by y. One may thus consider the ***maximum a posteriori (MAP) estimator***:

$$\arg\max_\theta \ell(\theta\,|\,y)\pi(\theta).$$

The MAP estimator is associated with the 0-1 losses presented in the previous paragraph. In continuous settings, one has

$$\int_\Theta \mathbf{1}_{\delta\neq\theta}\pi(\theta\,|\,y)d\theta=1,$$

and the 0-1 loss must be replaced by a sequence of losses

$$L_\varepsilon(\theta,a)=\mathbf{1}_{\|\theta-a\|>\varepsilon}.$$

The MAP estimate is then the limit of the Bayes estimates associated with L_ε when ε goes to 0. The MAP estimate can also be associated with a sequence of L_p losses where

$$L_p(\theta,a)=\|\theta-a\|^p.$$

It is in principle a penalized maximum likelihood (ML) estimator. Under a few regularity conditions on f and π, the asymptotic optimality properties of the regular ML estimator like consistency and efficiency are preserved for these Bayesian extension. As the sample size grows to infinity, the information contained in this sample becomes predominant compared to the fixed information brought by the prior π. Therefore, the MAP estimators are asymptotically equivalent to the ML estimators. Nevertheless, the MAP estimators have the advantage to be available also for finite sample sizes. But one should be aware that the MAP estimator is not always appropriate.

Example (taken from Robert [2001])**:** Consider

$$f(y\,|\,\theta)=\frac{1}{\pi}[1+(y-\theta)^2]^{-1}\qquad\text{and}\qquad\pi(\theta)=\frac{1}{2}e^{-|\theta|}.$$

The MAP estimator of θ is then always $\delta^*(y)=0$.

Region estimation

To summarize the inferential content of the posterior distribution, it is often convenient to be able to provide regions $C \subseteq \Theta$ containing a prescribed percentage of the probability mass of the posterior (Bernardo [2003]). For any $0 < q < 1$, we call each $C_q \subseteq \Theta$ a posterior ***q-credible region*** of θ if

$$\int_{C_q} \pi(\theta \,|\, y) d\theta = q,$$

i.e. if, given the data y, the parameter θ belongs to C_q with probability q. This definition reflects thus directly the intuitive understanding of a „confidence region“, in sharp contrast to the frequentist confidence intervals. A q-credible region is invariant under reparameterizations: if ϕ is a one-to-one transformation of θ, then $\phi(C_q)$ is a q-credible region with respect to $\phi(\theta)$ if C_q is a q-credible region with respect to θ.

There are generally infinitely many q-credible regions for each posterior, even in one dimension and even if C_q is an interval. One therefore wants to add further constraints: a q-credible region of minimum size (volume) is called ***highest probability density (HPD) region***, where all points inside the region have higher probability density than the points outside of the region. HPD regions are not reparameterization invariant. In one dimension, one therefore may prefer regions (intervals) derived by posterior quantiles: If θ_q is the $100q\%$ posterior quantile of θ, then

$$C_q^l = \left\{ \theta \;\middle|\; \theta \leq \theta_q \right\}$$

is a one-sided reparameterization invariant q-credible region, as well as the ***probability centred q-credible region*** given by

$$C_q^c = \left\{ \theta \;\middle|\; \theta_{(1-q)/2} \leq \theta \leq \theta_{(1+q)/2} \right\}.$$

For multi-modal posteriors, this definition may not be feasible, and an equivalent in higher dimensions is not easily found.

An alternative may be the following construction: Let $L(\theta, \tilde{\theta})$ be a loss function, and

$$\rho(\pi, \tilde{\theta} \,|\, y) = \mathbf{E}^{\pi}[L(\theta, a) \,|\, y] = \int_{\Theta} L(\theta, \tilde{\theta}) \pi(\theta \,|\, y) d\theta.$$

the posterior expected loss. Then a q-credible region C_q^* is called ***lowest expected loss (LEL) region***, if for all $\theta_1 \in C_q^*$ and all $\theta_2 \notin C_q^*$, it holds that

$$\rho(\pi, \theta_1 \,|\, y) < \rho(\pi, \theta_2 \,|\, y).$$

If the loss function is invariant under reparameterizations, the LEL credible region will also be invariant. Bernardo [2003] recommends especially the ***intrinsic credible region*** for general use, which is obtained as LEL credible region if an intrinsic loss is used: the intrinsic credible region is reparameterization invariant and definable in any dimension.

3.2.3 Justifications for Bayesian inference

Some connections and differences between the frequentist and the Bayesian viewpoints have already been exposed in the previous sections, with respect to decision rules and loss functions. This concerns mathematics. Nevertheless, the differences reach further, concerning not only mathematics but also the interpretation of probability theory, and thus the connection between mathematics and reality. In the present section, we try to clarify these differences in the interpretations.

Frequency versus Bayesian reasoning

As mentioned in the introduction, probability theory provides the connection between mathematical theory and real applications. The main two interpretations of probability, frequency probability and Bayesian probability, have caused many quarrels between, and polemics from their respective representatives. The main reason is that with Bayesian probability, many things are allowed which are strictly forbidden in the frequentist view. These regulations frequentist probability opposes reach also into the language we use when we speak about stochastic results. In the frequentist interpretation, "probabilities" are only and exclusively defined for frequencies of the outcomes of "random experiments" which can be repeated "independently" and principally infinitely often. Any other object or quantity which is not the frequency of a random experiment, or any individual observation, cannot have a probability. In contrast, in the Bayesian view, actually anything can be equipped with a distribution and thus with a probability. The notion "random" remains undefined and is just used to express that the considered objects vary in a more or less unpredictable way. Probabilities express the believes or certainty about (individual) events, or the plausibility for (individual) statements. We want to exploit the differences with some examples (see e.g. Jaynes [1976]).

Let θ be a "real world parameter", e.g. a natural constant, the expected number of heads of a coin thrown, or the size of a man of some given age. The frequentist viewpoint is:

- There exists a true value θ, and all possible outcomes of this value are equally well justified.
- Measured values of θ are disturbed by noise, but if θ is measured frequently, the mean value of the measurements goes near θ.
- Probabilities are distributions of these frequencies.

The Bayesian viewpoint is:

- θ varies "randomly" (randomly actually undefined; θ just varies in a more or less unpredictable way, unpredictable because one does not know better).
- One measures the distribution of θ and adjusts thus one's state of knowledge.
- Probabilities are really "distributions" of the value θ.

It is intuitively clear that there is actually no such thing as a "true size" of a man of some age. Thus, frequentist reasoning is actually not applicable in this case. The Bayesian interpretation still applies.

It must be emphasized that the construction of confidence intervals in the frequentist sense has to be seen as a random experiment. Thus, confidence intervals do not account for individual outcomes, and confidence intervals do not express the probability that a parameter θ lies inside this interval. Instead: If we have a 95% confidence interval, this means: if we observe θ and construct a confidence interval with these observed values, and if we do this infinitely often, then the (true) parameter θ lies inside this interval in 95% of the experiments; so, if we construct the 95% interval 100 times, then we should expect that in 5 times the true parameter θ does not lie inside this interval. This is against intuition, and is often misunderstood in practice. It is often more important to know how "confident" the individual or actual interval is than to know the outcome when the experiment is repeated infinitely often.

Similar arguments apply for hypothesis tests. If we test a hypothesis $H_0 = 0$ against $H_1 \neq 0$, and we say we reject this hypothesis on the 95% level, this means the rejection is false in 5% of the times we construct this test. Many other examples of the frequentist view which are counterintuitive or even yield wrong results can be found in Jaynes [1976]. We also have seen in the previous section that Bayesian decisions are often related to Frequentist decisions: they either coincide, or they yield better results.

Justification based on basic principles

One possibility to justify probability theory (and especially Bayesian probability theory) is to show that it agrees with certain basic principles. This is the direction followed e.g. in Robert [2001] and Berger [1980]. One of these basic principles is the ***Likelihood Principle*** which is attributed to Fisher (1959) or Barnard (1949) and formalized by Birnbaum (1962) (Robert [2001]):

> The information brought by an observation y about θ is entirely contained in the likelihood function $\ell(\theta \,|\, y)$. Moreover, if y_1 and y_2 are two observations depending on the same parameter θ, such that there exists a constant c satisfying
>
> $$\ell_1(\theta \,|\, y_1) = c\ell_2(\theta \,|\, y_2)$$
>
> for every θ, they then bring the same information about θ and must lead to identical inferences.

It should be noted that the Likelihood Principle is not identical to the Maximum Likelihood estimation method. Frequentist probability does not agree with the Likelihood Principle, whereas the pure Bayesian approach does; the introduction of loss functions in turn violates the Likelihood Principle. The introduction of loss functions is according to Berger [1980] due to a lack of time: If we had an infinite amount of time, we were able to determine the prior $\pi(\theta)$ exactly and use a pure Bayesian analysis satisfying the Likelihood Principle; with only a finite amount of time, it is necessary to approximate prior beliefs.

For further discussions of this and other principles see Robert [2001] and Berger [1980].

Cox's theorem

Cox [1946] claimed that probability (after a suitable rescaling) is the only reasonable way for dealing with uncertainty, plausibility or similar concepts of impreciseness.

This statement cannot be proven in a mathematical sense, because it is not a mathematical theorem. It has to be made plausible by common sense reasoning. To accomplish this, Cox formulated some requirements he thought necessary for a good calculus of plausibility of statements. Based on these requirements, he claimed mathematically that probability is inevitably the only possible model if these requirements are accepted. In his posthumous book [Jaynes, 2003], Jaynes uses Cox's theorem as a cornerstone of his justification of Bayesianism. He states Cox's requirements as follows (according to Arnborg and Sjödin [2000] and Arnborg and Sjödin [2001]):

(I) Divisibility and comparability: The plausibility of a statement is a real number and dependent on information we have related to the statement;

(II) Consistency: If the plausibility of the statement can be derived in two ways, the two results must be equal;

(III) Common sense: Plausibilities should vary sensibly with the assessment of plausibilities; deductive propositional logic should be the special case of reasoning with statements known to be true or to be false in the model.

Especially the common sense requirement is in itself rather imprecise and thus open to controversies. Furthermore, neither Cox nor Jaynes are very rigorous in their mathematical derivations and assume (sometimes only implicitly) additional strong requirements.

We introduce the following notation (Arnborg and Sjödin [2000], Arnborg and Sjödin [2001]): Denote by $pl(A|C)$ or short $A|C$ the plausibility of a statement A given that we know C to be true (thus, $A|C$ does not denote a statement but a real number). Cox introduces the function F defining the plausibility $A \wedge B|C$ (or short $AB|C$) of the conjunction A and B given C to be true,

$$AB|C = F(A|BC, B|C),$$

and the function S defining the plausibility $\overline{A}|C$ of the negation of A given C to be true,

$$\overline{A}|C = S(A|C).$$

Cox uses then some strong regularity conditions on F and S, e.g. associativity and twofold differentiability for F, to be able to proof that there must be a strictly monotone scaling w of the plausibility measure that satisfies the rules of probability:

$$w(F(x,y)) = w(x)w(y), \qquad w(S(x)) = 1 - w(x),$$

i.e. F is multiplication and $S(x)$ is $1-x$ after scaling with w.

Aczél [1966] releases the differentiability condition on F and introduces the partial function G defining the plausibility $A \vee B|C$ of the disjunction A or B given C to be true,

$$A \vee B|C = G(A|C, B\overline{A}|C).$$

He still needs a continuity assumption on G.

The continuity assumptions on F (and G) are needed to extend associativity to the whole domain of real numbers. Paris [1994] releases the continuity requirement of F, but he replaces it by another density requirement. Halpern [1999a] gives a "counterexample" for a finite model (where the density requirements do not apply), where F is not associative and thus not extendable to multiplication. As Snow [1998] points out, Halpern's example is not a counterexample to Cox's original statement because Cox *requires* F to be associative (see also Halpern's answer Halpern [1999b]).

Arnborg and Sjödin (Arnborg and Sjödin [2000], Arnborg and Sjödin [2001], Arnborg and Sjödin [2003]) use some new common sense requirements. Their new requirements replacing the density assumptions is based on refinability of a model: In any model, one could wish to be able to refine it by adding some new statements to the model, and if this is done in a reasonable way, it should never lead to inconsistencies. This is the requirement of the ***Refinability assumption*** (Arnborg and Sjödin [2000]): In a plausibility model with a conditional event of plausibility p, it must be possible to introduce a new subcase B of a non-false event A with plausibility value p given to $B|A$. In particular, it should be possible to define new subcases B and B' of a non-false event A such that they are ***information independent***, i.e. $B|B'A = B|A$ and $B'|BA = B'|A$. Information independence means that knowledge of the plausibility of one subcase does not affect the plausibility of the other. Arnborg and Sjödin add also the requirement of the ***Monotonicity assumption***: The domain of plausibility is ordered, S is strictly decreasing, and F and G are strictly increasing in each argument if the other argument is not $\perp$ (the smallest plausibility value). Moreover, $F(x,y) \leq x$ and $G(x,y) \geq x$. As is noted in [Arnborg and Sjödin, 2003], if one weakens the strict monotinicity condition which is required for S and replaces it by the requirement that S is only non-increasing, this would lead to completely different conclusions (see de Brucq et al. [2002]). With these requirements, it follows that each finite plausibility model is rescalable to probability.

For infinite models, this is not possible without further assumptions, see the counterexample in Arnborg and Sjödin [2000]. From this it follows that the refinability requirement is really weaker than the usual density requirements. For infinite models, one first needs a ***closedness assumption***: The functions F, S and G can be extended to an ordered domain D such that

$$F : D \times D \longrightarrow D, \qquad S : D \longrightarrow D, \qquad G : E \longrightarrow D$$

with

$$E := \big\{(x,y) \in D \times D \;\big|\; x \leq S(y)\big\}.$$

Then, one either has to assume an additional separability assumption, or else one has to accept extended probabilities (Arnborg and Sjödin [2001]). Let

$$x^1 := x \qquad \text{and} \qquad x^n := F(x, x^{n-1}).$$

Then the ***separability assumption*** introduced as a weaker assumption than the continuity assumptions is the following:

- For every $x < y$ and c, there are n, m sucht that $x^n < c^m < y^n$.

With these assumptions, it follows that also infinite plausibility models are rescalable to probability. If one does not want to accept the separability assumption, one has to accept extended probabilities: An ***extended probability model*** is a model based on probabilities taking values in an ordered field generated by the reals and an ordered set of infinitesimals. An ***infinitesimal*** is a non-zero element which in absolute value is smaller than any positive real. Conways field **No** of surreal numbers is universal in the sense that every totally ordered field can be embedded into **No**. By replacing the real values with **No**, Arnborg and Sjödin [2001] can show that each plausibility model fulfilling the monotonicity, refinability, and closedness assumptions can be uniquely embedded in a minimal ordered field where, after rescaling, multiplication and addition are extensions of F and G, respectively.

One popular example concerning a logic for reasoning under uncertainty is Fuzzy Logic. If one accepts Cox's theorem this disqualifies Fuzzy Logic from being a valid logic, except in cases where it is equivalent to probability theory. Arnborg and Sjödin [2001] model fuzzyness (impreciseness) in a different way. Instead of introducing events A as (objective) fuzzy sets, one can introduce the judgements made by various experts and decision makers as $A|C_i$ for the judgement made by expert i, based on the information available to expert i. This makes plausibility and fuzziness orthogonal concepts, and the question arises how these different priors can be combined. This will in the end lead to (convex) sets of probability distributions as models for this kind of extended plausibility, called Robust (Extended) Bayesian Analysis (see also the last paragraph in subsection 3.3.1).

Exchangeability and representation theorems Another argument for the inevitability of the use of probabilities to describe uncertainties is given by the notion of exchangeability and the corresponding representation theorem (see Bernardo [in press]). We call a set of random vectors $\{x_1,\ldots,x_n\}$, $x_j \in \mathscr{X}$, ***exchangeable*** if their joint distribution is invariant under permutations. An infinite sequence of random vectors is exchangeable if all its finite subsequences are exchangeable. In particular, any i.i.d. random sample from any model is exchangeable (only the values of a sample $\{x_1,\ldots,x_n\}$ matters, not their order). The general ***representation theorem*** implies that, if a set of observations is assumed to be a subset of an exchangeable sequence, then it constitutes a random sample from a probability model $\{p(x|\omega),\omega\in\Omega\}$, described in terms of some parameter vector ω; furthermore this ω is defined as the limit (as $n\to\infty$) of some function of the observations, and available information about the value of ω must necessarily be described by some probability distribution $p(\omega)$. This formulation includes "nonparametric" (distribution free) modelling, where ω may index, for instance, all continuous probability distributions on $\mathscr{X}$ (the collection of cumulative density functions has the power of the continuum and can thus be indexed by $\mathbb{R}$, see e.g. Robert [2001], ex. 1.2). Under exchangeability, and therefore under any assumption of random sampling, the general representation theorem provides an existence theorem for a probability distribution $p(\omega)$ on the parameter space Ω, and this argument depends only on mathematical probability theory.

3.3 Priors

The prior distribution is the key to Bayesian inference (Robert [2001]). Its determination is thus the most important step in drawing this inference. In practice, the available information is seldom precise enough to lead to an exact determination of the prior distribution. There is no such thing as *the* prior distribution. The prior should rather be seen as a tool summarizing available information as well as uncertainty related with this information. Ungrounded prior distributions lead to unjustified posterior inference: it is always possible to choose a prior distribution that gives the answer one wishes. The prior determination is therefore the most critical and most criticized point of Bayesian analysis.

3.3.1 Strategies for prior determination

We follow once more Robert [2001] and Berger [1980].

The possibilities for prior determination may be divided into three categories:

- Subjective priors,
- Conjugate priors,
- Objective (non-informative) priors.

Subjective prior determination

Some possibilities are:

- Use some prior knowledge about θ and approach the prior π e.g. by a histogram.
- Use empirical or hierarchical Bayes methods. We will describe these Bayes methods in more detail later in section 3.3.2.
- Select a ***maximum entropy prior*** (Jaynes [1980], Jaynes [1983]) if prior characteristics (moments, quantiles) are known:

  $$\mathbf{E}^{\pi}[g_k(\theta)] \qquad \text{for } k = 1, \ldots, K.$$

  The prior is based on the ***entropy*** introduced for the finite case by Shannon [1948] as a measure of uncertainty in information theory and signal processing:

  $$\mathscr{E}(\pi) = -\sum_i \pi(\theta_i) \log(\pi(\theta_i))$$

  for

  $$\Theta = \{\theta_1, \ldots, \theta_n\}.$$

  In the continuous case a reference measure π_0 has to be chosen:

  $$\mathscr{E}(\pi) = \mathbf{E}^{\pi_0}\left[\log\left(\frac{\pi(\theta)}{\pi_0(\theta)}\right)\right] = \int_{\theta\in\Theta} \log\left(\frac{\pi(\theta)}{\pi_0(\theta)}\right) \pi_0(d\theta),$$

being also the Kullback-Leibler divergence of π from π_0. The maximum entropy prior maximizes the entropy in the information-theoretical sense, that is, it minimizes the information brought through π about θ, and is given in the discrete resp. continuous case as

$$\pi^*(\theta_i) = \frac{\exp\left(\sum_{k=1}^K \lambda_k g_k(\theta_i)\right)}{\sum_j \exp\left(\sum_{k=1}^K \lambda_k g_k(\theta_j)\right)}, \qquad \pi^*(\theta) = \frac{\exp\left(\sum_{k=1}^K \lambda_k g_k(\theta)\right)\pi_0(\theta)}{\int_\Theta \exp\left(\sum_{k=1}^K \lambda_k g_k(\tilde{\theta})\right)\pi_0(d\tilde{\theta})},$$

the λ_k's being derived from the contraints $\mathbf{E}^\pi[g_k(\theta)]$ as Lagrange multipliers. A problem is the choice of π_0: it is seen as the completely noninformative distribution. When a group structure is available, it is usually taken to be the associated right-invariant Haar measure.

- Parametric approximations: Restrict the choice of π to parameterized densities $\pi(\theta\,|\,\lambda)$ and determine the hyperparameters λ through the moments or quantiles of π (the latter being more robust).

Conjugate priors

Definition 3.7 (Raiffa and Schlaifer [1961])**:** *A family $\mathscr{F}$ of probability distributions on Θ is **conjugate for a likelihood function** $f(y\,|\,\theta)$ if, for every $\pi \in \mathscr{F}$, the posterior distribution $\pi(\theta\,|\,y)$ also belongs to $\mathscr{F}$.*

Conjugate priors are only of interest in the case where $\mathscr{F}$ is parameterized. The choice of a conjugate prior is mainly based on computational deliberations, because switching from prior to posterior distribution is reduced to an updating of the corresponding parameters. The computation of posterior densities is thus really tractable and simple. But they often are only first approximations to adequate priors.

Exponential family The most important examples of conjugate priors are given for exponential families (see e.g. Robert [2001]):

Definition 3.8: *Let μ be a σ-finite measure on $\mathscr{Y}$, and let Θ be the parameter space. Let C and h be functions, respectively, from Θ and $\mathscr{Y}$ to $\mathbb{R}_{\geq 0}$, and let R and T be functions from Θ and $\mathscr{Y}$ to $\mathbb{R}^k$. The family of distributions with densities (with respect to μ)*

$$f(y\,|\,\theta) = C(\theta)h(y)\exp\left(R(\theta)^\top T(y)\right)$$

*is called **exponential family** of dimension k. If $\Theta \subseteq \mathbb{R}^k$, $\mathscr{Y} \subseteq \mathbb{R}^k$, and*

$$f(y\,|\,\theta) = C(\theta)h(y)\exp(\theta^\top y),$$

*then the family is called **natural**.*

Examples of the exponential family are common distributions like Normal, Gamma, Chi-square, Beta, Dirichlet, Bernouilli, Binomial, Multinomial, Poisson, Negative Binomial, Geometric, Weibull, or Wishart distributions. Not belonging to the exponential family are the Cauchy, Uniform or Pareto distributions.

By a change of variables from y to $z = T(y)$ and a reparameterization from θ to $\eta = R(\theta)$, it is usually enough to consider the natural form. For a natural exponential family, let

$$N := \{\theta \mid \int_{\mathscr{Y}} h(y)\exp(\theta^\top y)d\mu(y) < +\infty\}$$

be the so-called ***natural parameter space***. The natural form can also be rewritten as

$$f(y\,|\,\theta) = h(y)\exp(\theta^\top y - \psi(\theta))$$

where $\psi(\theta)$ is called the ***cumulant generating function***, because (see e.g. Robert [2001]):

Theorem 3.5: *If $\theta \in \overset{o}{N}$ (the interior of the natural parameter space N), the cumulant generating function ψ is C^∞ and*

$$\mathbf{E}_\theta[y] = \nabla\psi(\theta), \qquad \mathrm{Cov}(y_i, y_j) = \frac{\partial^2\psi}{\partial\theta_i\partial\theta_j}(\theta),$$

where ∇ denotes the gradient operator.

Thus, one can fully understand the mean and covariance structure by differentiating ψ.

The exponential family has other interesting analytical properties: For any sample

$$y_1, \ldots, y_n \sim f(y\,|\,\theta)$$

there exists a sufficient statistic of *constant* dimension:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \in \mathbb{R}^k.$$

The converse is the Pitman-Koopman-Lemma (1936):

Theorem 3.6 (Pitman-Koopman Lemma)**:** *If a family of distributions $f(\cdot\,|\,\theta)$ is such that, for a sample size large enough, there exists a sufficient statistic of constant dimension, then the family is exponential if the support of $f(\cdot\,|\,\theta)$ does not depend on θ.*

(The restriction on the support of $f(y\,|\,\theta)$ is necessary for the lemma to hold because the uniform $\mathscr{U}([-\theta,\theta])$ and the Pareto $\mathscr{P}(\alpha,\theta)$ distributions also satisfy this property; for these also conjugate priors exist, although they do not belong to the exponential family.)

A conjugate prior family for a natural exponential family is given by

$$\pi(\theta\,|\,\mu,\lambda) = K(\mu,\lambda)\exp(\theta^\top\mu - \lambda\psi(\theta)),$$

where $K(\mu,\lambda)$ is the normalizing constant of the density. The corresponding posterior distribution is

$$\pi(\theta\,|\,\mu + y, \lambda + 1),$$

which is σ-finite, and induces a probability distribution on Θ if and only if

$$\lambda > 0 \qquad \text{and} \qquad \frac{\mu}{\lambda} \in \overset{o}{N},$$

$f(y\,\vert\,\theta)$	$\pi(\theta)$	$\pi(\theta\,\vert\,y)$	$\hat{\theta}=\delta^{\pi}(y)$
Normal $\mathcal{N}(\theta,\sigma^2)$	Normal $\mathcal{N}(\mu,\tau^2)$	Normal $\mathcal{N}\left(\frac{\sigma^2\mu+\tau^2 y}{\sigma^2+\tau^2},\frac{\sigma^2\tau^2}{\sigma^2+\tau^2}\right)$	$\frac{\sigma^2\mu+\tau^2 y}{\sigma^2+\tau^2}$
Poisson $\mathcal{P}(\theta)$	Gamma $\mathcal{G}(\alpha,\beta)$	Gamma $\mathcal{G}(\alpha+y,\beta+1)$	$\frac{\alpha+y}{\beta+1}$
Gamma $\mathcal{G}(\nu,\theta)$	Gamma $\mathcal{G}(\alpha,\beta)$	Gamma $\mathcal{G}(\alpha+\nu,\beta+y)$	$\frac{\alpha+\nu}{\beta+y}$
Binomial $\mathcal{B}(n,\theta)$	Beta $\mathcal{B}e(\alpha,\beta)$	Beta $\mathcal{B}e(\alpha+y,\beta+n-y)$	$\frac{\alpha+y}{\alpha+\beta+n}$
Negative Binomial $\mathcal{N}eg(n,\theta)$	Beta $\mathcal{B}e(\alpha,\beta)$	Beta $\mathcal{B}e(\alpha+n,\beta+y)$	$\frac{\alpha+n}{\alpha+\beta+y+n}$
Multinomial $\mathcal{M}_k(\theta_1,\ldots,\theta_k)$	Dirichlet $\mathcal{D}(\alpha_1,\ldots,\alpha_k)$	Dirichlet $\mathcal{D}(\alpha_1+y_1,\ldots,\alpha_k+y_k)$	$\frac{\alpha_i+y_i}{(\sum_j\alpha_j)+n}$
Normal $\mathcal{N}(\mu,1/\theta)$	Gamma $\mathcal{G}(\alpha/2,\beta/2)$	Gamma $\mathcal{G}(\alpha+1,\beta+(\mu-y)^2)$	$\frac{\alpha+1}{\beta_k+(\mu-y)^2}$

Table 3.1: Conjugate priors and posterior mean estimates

and only if this holds, $K(\mu,\lambda)$ is well-defined.

Conjugate priors for several likelihoods from exponential families and their associated posterior distributions as well as the corresponding estimates under quadratic loss (posterior mean) are summarized in table 3.1 (taken from Robert [2001]).

If Θ is an open set in $\mathbb{R}^k$ and θ has the prior distribution

$$\pi_{\lambda,\mu}(\theta)\propto\exp(\theta^{\top}\mu-\lambda\psi(\theta))$$

conjugate for a natural exponential family $f(y\,\vert\,\theta)$, and

$$\xi(\theta):=\mathbf{E}[f(y\,\vert\,\theta)]$$

is the expectation, then

$$\mathbf{E}^{\pi}[\xi(\theta)]=\mathbf{E}^{\pi}[\nabla\psi(\theta)]=\frac{\mu}{\lambda}.$$

Thus, if $y_1,\ldots,y_n$ are i.i.d. $f(y\,\vert\,\theta)$, then the posterior mean of $\xi(\theta)$ is linear in y:

$$\mathbf{E}^{\pi}[\xi(\theta)\,\vert\,y_1,\ldots,y_n]=\mathbf{E}^{\pi}[\nabla\psi(\theta)\,\vert\,y_1,\ldots,y_n]=\frac{\mu+n\overline{y}}{\lambda+n}.$$

This can be extended to the case where $\pi_{\lambda,\mu}$ is improper, for instance $\lambda=0$ and $\mu=0$. In this case, the posterior expectation is $\overline{y}$, which is also the maximum likelihood estimator of $\xi(\theta)$.

Noninformative priors

In the absence of prior information one wants the prior distributions to be solely derived from the sample distribution $f(y\,\vert\,\theta)$. These noninformative priors should be considered as reference or default priors. They do not represent total ignorance. We follow again Robert [2001] and Berger [1980].

Laplace's prior The first example of a noninformative prior is ***Laplace's prior***, based on the "Principle of insufficient reason". In the finite case, $\Theta = \{\theta_1, \ldots, \theta_n\}$, Laplace's prior is

$$\pi(\theta_i) = 1/n.$$

The extension to continuous spaces leads to the improper prior

$$\pi(\theta) \propto 1.$$

One problem with this prior is that the posterior densities may also be improper. Another problem is the lack of reparameterization invariance: if we switch from $\theta \in \Theta$ to $\eta = g(\theta)$ with a one-to-one transformation g, prior information is still totally missing and should not be modified. But, if $\pi(\theta) \equiv 1$, the corresponding prior distribution on η is

$$\pi^*(\eta) = \left| \frac{d}{d\eta} g^{-1}(\eta) \right|$$

which is usually not constant.

Invariant prior The invariance principle is to consider the deliberation that the properties of a statistical procedure should not depend on the unit of measurement, in a general sense:

- Independence of the scale leads to ***scale-invariant*** estimators.
- Independence of the origin leads to ***translation-invariant*** estimators.
- Independence of order of the observations leads to ***symmetric*** estimators.

In all cases, the invariance structure is given through group actions (Robert [2001] and Berger [1980]):

Definition 3.9 (Invariant decision problem)**:** *Let $(\mathscr{Y}, \Theta, f(y|\theta))$ be a parametric statistical problem, $\mathscr{A}$ a decision space, and $L : \Theta \times \mathscr{A} \longrightarrow \mathbb{R}_{\geq 0}$ a loss function. Let $\mathscr{G}$ be a group acting on $\mathscr{Y}$.*

*(i) The statistical model $f(y|\theta)$ is said to be **invariant under the action of the group** $\mathscr{G}$, if for every $g \in \mathscr{G}$, there exists a unique $\theta^* \in \Theta$ such that $g(y)$ is distributed according to the density $f(g(y)|\theta^*)$. We denote $\theta^* = \bar{g}(\theta)$.*

*(ii) If the model is invariant under the action of $\mathscr{G}$, the loss L is said to be **invariant under the action of the group** $\mathscr{G}$ if, for every $g \in \mathscr{G}$ and $a \in \mathscr{A}$, there exists a unique decision $a^* \in \mathscr{A}$, such that $L(\theta, a) = L(\bar{g}(\theta), a^*)$ for every $\theta \in \Theta$. We denote $a^* = \tilde{g}(a)$. The decision problem is said to be **invariant under** $\mathscr{G}$.*

We are thus concerned with the following three groups and their actions:

$$\begin{aligned} \mathscr{G} &: y \mapsto g(y) \sim f(g(y)|\bar{g}(\theta)), \\ \overline{\mathscr{G}} &: \theta \mapsto \bar{g}(\theta), \\ \widetilde{\mathscr{G}} &: L(\theta, a) = L(\bar{g}(\theta), \tilde{g}(a)). \end{aligned}$$

One then restricts the class of decision rules to the ***invariant*** or ***equivariant decision rules***, i.e. those satisfying

$$\delta(g(y)) = \tilde{g}(\delta(y)).$$

One has to determine a prior which is ***invariant under the action of the group*** $\tilde{\mathscr{G}}$:

$$\pi^*(\bar{g}(A)) = \pi^*(A)$$

for every measurable subset A of Θ and every $g \in \mathscr{G}$. The solution is given by the right Haar measure (see e.g. Robert [2001]):

Theorem 3.7: *The best equivariant decision rule for θ is the Bayes decision rule δ^{π^*} associated with the right Haar measure π^* on Θ, and the corresponding invariant loss.*

This is in most cases an improper prior, because invariant probability distributions are rare, since they can only exist for compact groups $\tilde{G}$. We provide some examples from Robert [2001]:

Examples: Let $f(y)$ for $y \in \mathbb{R}^d$ be a probability density.

(a) The model family

$$\mathscr{M}_1 = \left\{ f(y-\theta), \theta \in \mathbb{R}^d \right\}$$

is said to be parameterized by the ***location parameter*** θ. The model class is translation-invariant, i.e. for $\tilde{y} = y - y_0$ with $y_0 \in \mathbb{R}^d$ it follows

$$f(\tilde{y} - (\theta - y_0)) \in \mathscr{M}_1,$$

and the invariance principle requires that the prior should be translation-invariant, too, i.e.

$$\pi^*(\theta) = \pi^*(\theta - \theta_0) \qquad \text{for all } \theta_0 \in \mathbb{R}^d.$$

The solution is $\pi^*(\theta) = c$ for a constant $c \in \mathbb{R}$.

(b) The model family

$$\mathscr{M}_2 = \left\{ 1/\sigma f(y/\sigma), \theta \in \mathbb{R}_{>0} \right\}$$

is said to be parameterized by a ***scale parameter*** $\sigma > 0$. The model class is scale-invariant, i.e. for $\tilde{y} = y/s$ with $s \in \mathbb{R}_{>0}$ it follows

$$1/(\sigma/s) f(\tilde{y}/(\sigma/s)) \in \mathscr{M}_2,$$

and the invariance principle requires that the prior should be scale-invariant, too, i.e.

$$\pi^*(\sigma) = \frac{1}{c}\pi^*(\sigma/c).$$

This implies $\pi^*(\sigma) = \alpha/\sigma$ for a constant $\alpha \in \mathbb{R}$.

Disadvantages of this approach are that the determination of invariant priors requires the invariance to be part of the decision problem. This often leads to ambiguities, since it is sometimes possible to consider several invariant groups. The natural invariance structure can also be either too weak or too strong to lead to good estimators. Such natural invariance structures are even missing in most discrete setups (e.g. Poisson).

Jeffreys' prior Jeffreys (Jeffreys [1946], Jeffreys [1961]) proposed an approach which avoids the need to take a natural invariance structure into account. Jeffreys' prior is based on the Fisher information matrix $I(\theta)$ defined through the components

$$I_{ij}(\theta) = -\mathbf{E}_\theta\left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log f(y\,|\,\theta)\right] \qquad \text{for } i,j = 1,\ldots,d.$$

The Fisher information is intended to be an indicator of the amount of information brought by the model (or the observation) about θ (Fisher 1956). ***Jeffreys' noninformative prior*** is then defined as:

$$\pi^J(\theta) \propto |\det I(\theta)|^{1/2}.$$

In the one-dimensional case, Jeffreys' prior agrees with most invariant priors, e.g. with the location- and scale-invariant priors given in the examples of the previous paragraph. Furthermore, it is parameterization invariant. But in higher dimensions, Jeffreys' approach may lead to incoherences or even paradoxes, and it is not coherent for the likelihood principle. Nevertheless, this method provides one of the best automated techniques to derive noninformative prior distributions.

Reference priors The concept of reference priors (Bernardo [1979], Berger and Bernardo [1989]) generalizes Jeffreys' priors by distinguishing between nuisance and interest parameters. The principle is to eliminate the nuisance parameters by using Jeffreys' prior where the parameter of interest remains fixed. When $y \sim f(y\,|\,\theta)$ and $\theta = (\theta_1, \theta_2)$, where θ_1 is the parameter of interest, the ***reference prior*** is obtained by first defining $\pi^J(\theta_2\,|\,\theta_1)$ as Jeffreys' prior associated with $f(y\,|\,\theta)$ when θ_1 is fixed, then deriving the marginal distribution

$$\tilde{f}(y\,|\,\theta_1) = \int f(y\,|\,\theta_1,\theta_2)\pi^J(\theta_2\,|\,\theta_1)d\theta_2,$$

and computing Jeffreys' prior $\pi^J(\theta_1)$ associated with $\tilde{f}(y\,|\,\theta_1)$. Often, the marginalization integral is not defined. Berger and Bernardo (1989) then suggest to derive the reference prior for compact subsets Θ_n of Θ and to consider the limit of the resulting reference priors π_n as $n \to \infty$ and Θ_n goes to Θ. In general, the resulting limit does not depend on the choice of the sequence of compact subsets.

Reference priors depend on the way parameters are ordered, an advantage compared to Jeffreys' prior because nuisance parameters are considered in a different way. When no ordering of the parameters are given, Berger and Bernardo [1992] suggest that one considers as a noninformative prior the reference prior for which each component of θ is considered separately; in contrast, Jeffreys' prior treats θ as single group of parameters.

Matching priors ***Matching priors*** are noninformative priors that match with good frequentist properties, i.e. properties that hold on the average in y rather than conditional on y. A common approach is to require that some posterior probabilities must asymptotically coincide with the corresponding frequentist coverage probability. Besides the technical difficulty one faces in handling matching priors, there is a conceptual difficulty in asking for frequentist

coverage probability when constructing a prior distribution; the goal is to condition on the observation rather than to rely on frequentist long-term properties. It also violates the Likelihood Principle. Robert [2001] does not recommend this method.

Other approaches to noninformative priors Other approaches to non-informative priors are (see Robert [2001]):

- Rissanen's transmission information theory and minimum lengths priors;
- testing priors;
- stochastic complexity.

Problem of prior elicitation: Information fusion

The prior information is rarely rich enough to define a prior distribution exactly. Thus, this uncertainty must be included into the Bayesian model. Possibilities are:

- Further prior modelling,
- Upper and lower probabilities (Dempster-Shafer, DS, Modified DS),
- Imprecise probabilities (Walley).

Only for finite problems with some symmetry properties unique priors could be found, and therefore there is an inherent subjectivity in the choice of the prior. Different epxerts may choose different priors. Thus one concentrates on the fusion of different sources, for example different experts or several sensors. The field which investigates these issues is called Information Fusion. One possibility and an alternative to the above mentioned theories like the Dempster-Shafer theory is Robust Bayesianism, see Arnborg [2006]: Uncertainty and imprecision are orthogonal concepts, and priors are defined by a convex set of probabilities. Arnborg claims that the multiplication operator in Bayes' rule extends to the prior sets (all probabilities in the first convex set are multiplied by all probabilities in the second convex set).

3.3.2 Hierarchical Bayes

Another approach to include uncertainty about prior distribution into the Bayesian model is given by hierarchical models. It is based on a decomposition of the prior distribution into several conditional levels of distributions, mostly two levels: The first level often is a conjugate prior, with parameters distributed according to the second-level distribution. There are also real live motivations for such decompositions (e.g. multiple experiments, meta-analysis). We follow Robert [2001].

Definition 3.10: *A **Hierarchical Bayes model** is a Bayesian statistical model*

$$(f(y\,|\,\theta),\pi(\theta))$$

where

$$\pi(\theta)=\int_{\Theta_1\times\cdots\times\Theta_n}\pi_1(\theta\,|\,\theta_1)\pi_2(\theta_1\,|\,\theta_2)\cdots\pi_{n+1}(\theta_n)d\theta_1\cdots d\theta_n.$$

*The parameters θ_i are called **hyperparameters** of level i, $1\le i\le n$.*

The use of hierarchical models allows often the separation of structural information from subjective information. In non-informative settings, we can see it as a compromise between Jeffreys' non-informative distributions and conjugate distributions. Uncertainties are pushed a step further away, which leads to a robustification of the Bayesian procedure.

Furthermore, hierarchical models often simplify Bayesian calculations, e.g. easy decomposition of the posterior distribution. For instance, if

$$\theta\,|\,\theta_1\sim\pi_1(\theta\,|\,\theta_1),\qquad \theta_1\sim\pi_2(\theta_1),$$

then the posterior distribution of θ is

$$\pi(\theta\,|\,y)=\int_{\Theta_1}\pi(\theta\,|\,\theta_1,y)\pi(\theta_1\,|\,y)d\theta_1$$

where

$$\pi(\theta\,|\,\theta_1,y)=\frac{f(y\,|\,\theta)\pi_1(\theta\,|\,\theta_1)}{m_1(y\,|\,\theta_1)},\qquad m_1(y\,|\,\theta_1)=\int_\Theta f(y\,|\,\theta)\pi_1(\theta\,|\,\theta_1)d\theta,$$

$$\pi(\theta_1\,|\,y)=\frac{m_1(y\,|\,\theta_1)\pi_2(\theta_1)}{m(y)},\qquad m(y)=\int_{\Theta_1}m_1(y\,|\,\theta_1)\pi_2(\theta_1)d\theta_1.$$

Moreover, the decomposition works for the posterior moments, that is, for every suitable functions h:

$$\mathbf{E}^\pi[h(\theta)\,|\,y]=\mathbf{E}^{\pi(\theta_1\,|\,y)}[\mathbf{E}^{\pi_1}[h(\theta)\,|\,\theta_1,y]]$$

where

$$\mathbf{E}^{\pi_1}[h(\theta)\,|\,\theta_1,y]=\int_\Theta h(\theta)\pi(\theta\,|\,\theta_1,y)d\theta.$$

For the hierarchical model

$$\pi(\theta)=\int_{\Theta_1\times\cdots\times\Theta_n}\pi_1(\theta\,|\,\theta_1)\pi_2(\theta_1\,|\,\theta_2)\cdots\pi_{n+1}(\theta_n)d\theta_1\cdots d\theta_n,$$

the ***full conditional distribution*** of θ_i given x and the θ_j's, $j\neq i$, i.e.

$$\pi(\theta_i\,|\,y,\theta,\theta_1,\ldots,\theta_n),$$

satisfies the following local conditioning property:

$$\pi(\theta_i\,|\,y,\theta,\theta_1,\ldots,\theta_n)=\pi(\theta_i\,|\,\theta_{i-1},\theta_{i+1}),$$

with the convention $\theta_0=\theta$ and $\theta_{n+1}=0$.

Nevertheless, we rarely are provided with explicit derivations of corresponding Bayes estimators. The natural solution in hierarchical settings is to use a simulation based approach exploiting the hierarchical conditional structure, such as the Gibbs sampler (see section 3.5).

Example (Hierarchical extension for the normal model (Robert [2001]))**:** For the normal model

$$y \sim \mathcal{N}_d(\theta, \Sigma)$$

with the first level conjugate distribution $\pi_1(\theta \,|\, \mu, \Sigma_\pi)$ given by

$$\theta \sim \mathcal{N}_d(\mu, \Sigma_\pi)$$

and the second level prior $\pi_2(\mu, \Sigma_\pi)$, the hierarchical Bayes estimator is

$$\delta^\pi(y) = \mathbf{E}^{\pi_2(\mu,\Sigma_\pi\,|\,y)}[\delta(y \,|\, \mu, \Sigma_\pi)]$$

with

$$\begin{aligned} \delta(y \,|\, \mu, \Sigma_\pi) &= y - \Sigma W(y-\mu), \qquad & W &= (\Sigma + \Sigma_\pi)^{-1}, \\ \pi_2(\mu, \Sigma_\pi \,|\, y) &\propto (\det W)^{1/2} \exp\big(-(y-\mu)^\top W(y-\mu)/2\big)\pi_2(\mu, \Sigma_\pi). \end{aligned}$$

Empirical Bayes

As Robert [2001], p.478, puts it, "the appelation *empirical Bayes* is doubly defective because firstly, the method is not Bayesian and, secondly, genuine Bayesian methods are empirical, since they are based on data!" The method does not follow from the Bayesian principles since it approximates the prior distribution by frequentist methods when the prior information is too vague. It can be viewed as a dual method to the hierarchical Bayes analysis and is asymptotically equivalent to the Bayesian approach. It may be an acceptable approximation in problems for which a genuine Bayes modelling is too complicated or too costly. But it should be said that with increasing computational power and the development of MCMC methods, the need for empirical approximations to more complex hierarchical analyses diminishes.

Nonparametric empirical Bayes The ***empirical Bayes*** perspective as introduced by Robbins [1951] may be stated as follows: Given $(n+1)$ independent observations $y_1, \ldots, y_{n+1}$ with densities $f(y_i \,|\, \theta_i)$, the problem is to draw an inference on θ_{n+1}, under the additional assumption that the θ_i's have all been generated according to the same unknown prior distribution g. From a Bayesain point of view, this means that the sampling distribution is known, but the prior distribution is not. The marginal distribution,

$$f_g(y) = \int f(y \,|\, \theta) g(\theta) d\theta,$$

can then be used to recover the distribution g from the observations, since $y_1, \ldots, y_n$ can be considered as an i.i.d. sample from f_g. Deriving an approximation $\hat{g}_n$ in this manner, we can use it as a substitute for the true prior distribution, and propose the plug-in approximation to the posterior distribution

$$\tilde{\pi}(\theta_{n+1} \,|\, x_{n+1}) \propto f(x_{n+1} \,|\, \theta_{n+1}) \hat{g}_n(\theta_{n+1}).$$

This derivation is not Bayesian! A Bayesian approach, arguing from the ignorance of g, would index this distribution by a hyperparameter λ and would thus represent ignorance by a second-level prior distribution, $\pi_2(\lambda)$.

The empirical Bayes approach is problematical in many respects, see e.g. Robert [2001], section 10.4.

Parametric empirical Bayes The ***parametric empirical Bayes*** approach is a restricted version of nonparametric empirical Bayes. In exponential family settings, the prior distribution being unavailable, a simple choice is to take a conjugate prior $\pi(\theta\,|\,\lambda)$ associated with the sampling distribution $f(y|\,\theta)$. While the hierarchical approach introduces an additional distribution on the hyperparameters λ, the empirical Bayes analysis proposes to estimate these hyperparameters from the marginal distribution

$$m(y|\,\lambda) = \int_\Theta f(y|\,\theta)\pi(\theta\,|\,\lambda)d\theta$$

by some frequentist estimator $\hat{\lambda}(y)$, and to use $\pi(\theta\,|\,\hat{\lambda}(y),y)$ as a pseudo-posterior distribution. This method then appears as the parametric version of the original approach of Robbins [1956].

A defect with the empirical Bayes perspective is that it relies on frequentist methods to estimate the hyperparameters of $m(y|\,\lambda)$, although Bayesian techniques could be used as well. As estimators $\hat{\lambda}(y)$, a wide range of options is available: for instance, the estimator of λ can be derived by the moment method or the maximum likelihood method. The corresponding arbitrariness of empirical Bayes analysis is the major flaw of this theory. The most common approach is to use maximum likelihood estimators.

3.4 Stochastic models and Bayesian estimation

3.4.1 Static normal models

We follow Robert [2001]. Normal models are extensively used, in particular where the Central Limit Theorem approximation can be justified (econometrics, particle physics, etc.); it is often justified for asymptotic reasons. A d-dimensional normal model is given by a multivariate Gaussian distribution

$$\mathcal{N}_d(\theta,\Sigma)$$

with d-dimensional mean vector $\theta \in \mathbb{R}^d$ and symmetric positive definite $d \times d$ covariance matrix Σ. If Σ is known, the normal model together with a normal conjugate distribution $\theta \sim \mathcal{N}_d(\mu,A)$ yields the posterior distribution

$$\mathcal{N}_d(y-\Sigma(\Sigma+A)^{-1}(y-\mu),(\Sigma^{-1}+A^{-1})^{-1})$$

for the mean θ. Under quadratic loss, the Bayes estimator is then the posterior mean

$$\delta^\pi(y) = y-\Sigma(\Sigma+A)^{-1}(y-\mu) = (\Sigma^{-1}+A^{-1})^{-1}(\Sigma^{-1}y+A^{-1}\mu).$$

For repeated observations $y_1,\dots,y_n$ of the above normal model, the sufficient statistic

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} x_i \sim \mathcal{N}_d\Big(\theta, \frac{1}{n}\Sigma\Big)$$

directly extends this analysis (compare to subsection 3.3.1).

If the covariance matrix Σ is unknown, it is necessary to consider prior distributions on the parameter (θ,Σ). If Σ is known up to a multiplicative constant σ^2, it is usually possible to get back to the unidimensional case, i.e. when $y_1,\dots,y_n$ are i.i.d. $\mathcal{N}(\theta,\sigma^2)$. (The case where θ is known and σ^2 only is unknown is treated in subsection 3.3.1, see table 3.1). If we define the statistics

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad \text{and} \qquad s^2 = \sum_{i=1}^{n}(y_i-\bar{y})^2,$$

then the likelihood is

$$\ell(\theta,\sigma^2\,|\,\bar{y},s^2) \propto \sigma^{-n}\exp\left[-\frac{1}{2\sigma^2}\Big(s^2+n(\bar{y}-\theta)^2\Big)\right].$$

Considering first non-informative priors, Jeffreys' prior for this model is

$$\pi^*(\theta,\sigma) = \frac{1}{\sigma^2},$$

but for invariance reasons it is better to use the prior

$$\tilde{\pi}(\theta,\sigma) = \frac{1}{\sigma}.$$

In this case the posterior distribution of (θ,σ^2) associated with the prior $\tilde{\pi}$ is

$$\theta\,|\,\sigma^2,\bar{y},s^2 \sim \mathcal{N}\left(\bar{y},\frac{\sigma^2}{n}\right), \qquad \sigma^2\,|\,\bar{y},s^2 \sim \mathcal{IG}\left(\frac{n-1}{2},\frac{s^2}{2}\right)$$

(where $\mathcal{IG}$ denotes the inverse Gamma distribution).

Apart from the non-informative priors, it is also possible to derive conjugate priors, the conjugate posterior distributions having the same form as the posteriors in the non-informative setting. A pecularity occurring in this case is that θ and σ^2 are not a priori independent, i.e. the prior is of the form

$$\pi(\theta,\sigma) = \pi(\theta\,|\,\sigma^2)\pi(\sigma^2).$$

We consider only the general case where the parameters (θ,Σ) are totally unknown. Given observations $y_1,\dots,y_n$ of $\mathcal{N}_d(\theta,\Sigma)$, a sufficient statistic is

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad \text{and} \qquad S = \sum_{i=1}^{n}(y_i-\bar{y})(y_i-\bar{y})^\top,$$

and the likelihood given by

$$\ell(\theta,\Sigma\,|\,\bar{y},S) \propto |\Sigma|^{-n/2}\exp\left[-\frac{1}{2}\Big(n(\bar{y}-\theta)^\top\Sigma^{-1}(\bar{y}-\theta)+\operatorname{tr}(\Sigma^{-1}S)\Big)\right],$$

which suggests the following conjugate priors:

$$\theta \,|\, \Sigma \sim \mathcal{N}_d\left(\mu, \frac{\Sigma}{n_0}\right), \qquad \Sigma^{-1} \sim \mathcal{W}_d(\alpha, W)$$

where $\mathcal{W}_d$ denotes the Wishart distribution. The posterior distributions are then

$$\theta \,|\, \Sigma, \bar{y}, S \sim \mathcal{N}_d\left(\frac{n_0\mu + n\bar{y}}{n_0+n}, \frac{\Sigma}{n_0+n}\right), \qquad \Sigma^{-1} \,|\, \bar{y}, S \sim \mathcal{W}_d(\alpha+n, W_1(\bar{y}, S))$$

with

$$W_1(\bar{y}, S)^{-1} = W^{-1} + S + \frac{nn_0}{n+n_0}(\bar{y}-\mu)(\bar{y}-\mu)^\top.$$

A careful determination of the hyperparameters μ, n_0, α, W is required.

Linear normal models

We consider the usual regression model (see e.g. Robert [2001]):

$$y = X\theta + \varepsilon, \qquad \varepsilon \sim \mathcal{N}_k(0, \Sigma), \quad \theta \in \mathbb{R}^d$$

where X is the $k \times d$ ***regressor matrix***, and $y \sim \mathcal{N}_k(X\theta, \Sigma)$ is observed. If the covariance matrix Σ is known, this model can be analyzed in the same way as above when working conditional on X. A sufficient statistic is

$$\hat{\theta} = (X^\top \Sigma^{-1} X)^{-1} X^\top \Sigma^{-1} y$$

which is the maximum likelihood estimator and the least-squares estimator of θ. If the regressor matrix X is considered to be constant, conjugate distributions of the type

$$\theta \sim \mathcal{N}_d(A\beta, C), \qquad \text{where } \beta \in \mathbb{R}^q \ (q \le d)$$

may be considered; the inference is then made conditional on X, and A, C, or β may depend on X. If Σ is unknown, Jeffreys' prior is

$$\pi^J(\theta, \Sigma) = \frac{1}{|\Sigma|^{(k+1)/2}},$$

and the likelihood

$$\ell(\theta, \Sigma \,|\, y) \propto |\Sigma|^{-n/2} \exp\left(-\frac{1}{2}\operatorname{tr}\left[\Sigma^{-1}\sum_{i=1}^{n}(y_i - X_i\theta)(y_i - X_i\theta)^\top\right]\right).$$

This suggests the use of a Wishart distribution, but the posterior marginal distribution on θ is only defined for a sufficiently large sample size, and is not explicit for any sample size.

In the special case $\varepsilon \sim \mathcal{N}_k(0, \sigma^2 I_k)$, the least-squares estimator $\hat{\theta}$ has a normal distribution

$$\mathcal{N}_d(\theta, \sigma^2 (X^\top X)^{-1}).$$

Corresponding conjugate distributions on (θ, σ^2) are then

$$\theta \,|\, \sigma^2 \sim \mathcal{N}_d\left(\mu, \frac{\sigma^2}{n_0}(X^\top X)^{-1}\right), \qquad \sigma^2 \sim \mathcal{IG}\left(\frac{\nu}{2}, \frac{s_0^2}{2}\right)$$

since, if $s^2 = \|y - X\hat{\theta}\|_2^2$, the posterior distributions are

$$\theta \,|\, \hat{\theta}, s^2, \sigma^2 \sim \mathcal{N}_d\left(\frac{n_0\mu + \hat{\theta}}{n_0+1}, \frac{\sigma^2}{n_0+1}(X^\top X)^{-1}\right),$$
$$\sigma^2 \,|\, \hat{\theta}, s^2 \sim \mathcal{IG}\left(\frac{k-d+\nu}{2}, \frac{s^2 + s_0^2 + \frac{n_0}{n_0+1}(\mu-\hat{\theta})^\top X^\top X(\mu-\hat{\theta})}{2}\right).$$

3.4.2 Dynamic models

A ***dynamic model*** or ***time series model*** appears as a particular case of a parametric model where the distribution of the observed variables $y_1, \ldots, y_T$ varies over time:

$$f(y_1, \ldots, y_T \,|\, \theta) = \prod_{t=1}^{T} f_t(y_t \,|\, y_{1:t-1}, \theta)$$

where $y_{1:t-1}$ denotes the collection of previous variables $y_1, \ldots, y_{t-1}$ with the convention that $y_{1:0}$ is either empty or a fixed initial value y_0 (then belonging to the parameters θ). The inclusion of unobserved components in the variables y_t provides a fairly large sope of this model, including state space models. The dynamic models are more challenging than the static models because one usually requires stationarity constraints for them: A stochastic process (X_n) is called ***stationary***, if for any k, the marginal distribution of $(X_n, \ldots, X_{n+k})$ does not change if n varies (see e.g. Meyn and Tweedie [1993]). We follow Robert [2001].

Dynamical linear normal models

The AR(p) model The ***autoregressive model*** of order p, AR(p), is given by the dynamic model

$$y_t \sim \mathcal{N}\left(\mu - \sum_{i=1}^{p} \rho_i(y_{t-i} - \mu), \sigma^2\right)$$

or, equivalently,

$$y_t = \mu - \sum_{i=1}^{p} \rho_i(y_{t-i} - \mu) + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, \sigma^2).$$

In this model, the distribution of y_t only depends on a fixed number of past values, $y_{t-p:t-1}$, and the model can thus be expressed as a Markov chain when considering

$$z_t := y_{t:t-p+1}^\top = (y_t, y_{t-1}, \ldots, y_{t-p+1})^\top$$

since

$$z_t = \mu\mathbf{1} + B(z_{t-1} - \mu\mathbf{1}) + \varepsilon_t e_1,$$

where

$$\mathbf{1} = (1,\ldots,1)^\top, \qquad B = \begin{pmatrix} \rho_1 & \rho_2 & \cdots & \rho_p \\ 1 & 0 & \ldots & 0 \\ & & \ddots & \\ 0 & 0 & \ldots & 0 \end{pmatrix}, \qquad \text{and} \qquad e_1 = (1,0,\ldots,0)^\top.$$

The likelihood conditional on the negative-time values $y_0,\ldots,y_{-p+1}$ is given by:

$$\ell(\mu,\rho_1,\ldots,\rho_p,\sigma\,|y_{1:T},y_{0:(-p+1)}) = \\ \sigma^{-T}\prod_{t=1}^{T}\exp\left[-\left(y_t-\mu+\sum_{i=1}^{p}\rho_i(y_{t-i}-\mu)\right)^2/2\sigma^2\right].$$

A natural conjugate prior for the parameter $\theta = (\mu,\rho_1,\ldots,\rho_p,\sigma^2)$ is a normal distribution on $(\mu,\rho_1,\ldots,\rho_p)$ and an inverse gamma distribution on σ^2.

One may think about imposing stationarity constraints on the prior, given by restrictions on the values of the parameter $\theta := (\mu,\rho_1,\ldots,\rho_p)$ (see Robert [2001]):

Theorem 3.8: *The stochastic process defined by the AR(p) model with parameters $\theta := (\mu,\rho_1,\ldots,\rho_p)$ is stationary if and only if the roots of the polynomial*

$$P(y) = 1-\sum_{i=1}^{p}\rho_i y^i$$

are all outside the unit circle in the complex plane.

With the stationarity constraint, the parameters vary in a complicated subspace of the parameter space which is too difficult to deal with when $p > 3$. A solution is given by the Durbin-Levinson recursion (Monahan [1984]) which proposes a reparameterization of the parameters ρ_i to the ***partial autocorrelations***

$$\psi_i \in (-1,+1), \qquad i = 1,\ldots,p,$$

which allow for a uniform prior. The partial autocorrelations are also called ***reflection coefficients*** in the signal processing literature. They can be used to test stationarity, since, according to Schur's lemma, they must all be between -1 and $+1$ if the chain (y_t) is stationary. The following algorithm provides a constructive connection to deduce the parameters ρ_i from the coefficients ψ_i under the stationarity constraint:

0. Define $\varphi^{i,i} = \psi_i$ and $\varphi^{i,j} = \varphi^{i-1,j} - \psi_i\varphi^{i-1,i-j}$, for $i \geq 1$ and $j = 1,\ldots,i-1$.

1. Take $\rho_i = \varphi^{p,i}$ for $i = 1,\ldots,p$.

While the resulting prior (as well as the posterior) distribution on $(\rho_1,\ldots,\rho_p)$ is not explicit, this representation can be exploited for simulation purposes.

A different approach goes via the real and complex roots of the polynomial P, whose inverses are also within the unit circle (Huerta and West [1999]).

If one wants to use instead a non-informative prior, one possibility is the usual prior

$$\pi(\mu, \sigma^2, \rho) = 1/\sigma$$

(based on invariance considerations), or Jeffreys' prior. But Jeffreys' prior is controversal: Consider the AR(1) model with $\rho = \rho_1$. Jeffreys' prior associated with the stationary representation is

$$\pi_1^J(\mu, \sigma^2, \rho) \propto \frac{1}{\sigma^2} \frac{1}{\sqrt{1-\rho^2}}.$$

Within the non-stationary region $|\rho| > 1$, Jeffreys' prior is

$$\pi_2^J(\mu, \sigma^2, \rho) \propto \frac{1}{\sigma^2} \frac{1}{\sqrt{|1-\rho^2|}} \sqrt{\left|1 - \frac{1-\rho^{2T}}{T(1-\rho^2)}\right|}.$$

Thus, the dominant part of the prior is the non-stationary region, since it is equivalent to ρ^{2T}. Using the reference prior π_1^J for the whole region is not possible because this prior is only defined when the stationary constraint holds.

A proposed solution to this is to use a prior "symmetrized" to the region $|\rho| > 1$:

$$\pi^B(\mu, \sigma^2, \rho) \propto \frac{1}{\sigma^2} \begin{cases} 1/\sqrt{1-\rho^2} & \text{if } |\rho| < 1, \\ 1/|\rho|\sqrt{\rho^2-1} & \text{if } |\rho| > 1, \end{cases}$$

which has a more reasonable shape than the prior π_2^J (see Robert [2001] for references).

The MA(q) model The ***moving average model*** of order q, MA(q), is defined as

$$y_t = \mu + \varepsilon_t - \sum_{j=1}^{q} \vartheta_j \varepsilon_{t-j}, \qquad \varepsilon_t \sim \mathcal{N}(0, \sigma^2).$$

This is a special case of the ***Wold decomposition***

$$y_t = \mu + \varepsilon_t - \sum_{j=1}^{\infty} \vartheta_j \varepsilon_{t-j}, \qquad \varepsilon_t \sim \mathcal{N}(0, \sigma^2),$$

setting ψ_i equal to 0 for all $i < q$. For the MA(q) process, the autocovariances

$$\gamma_s = \mathrm{Cov}(y_t, y_{t+s})$$

are equal to 0 for all $|s| > q$, in contrast to e.g. the AR(1) model where the covariances between the terms are exponentially decreasing to 0 but remain always different from 0.

The MA(q) process is stationary whatever the parameters $(\vartheta_1, \ldots, \vartheta_q)$ are (this follows from the Wold decompositions), but for invertibility and identifiability considerations, the polynomial

$$Q(y) = 1 - \sum_{j=1}^{q} \vartheta_j y^j$$

must have all its roots outside the unit circle.

Example: For the MA(1) model

$$y_t = \mu + \varepsilon_t - \vartheta_1 \varepsilon_{t-1}$$

we get

$$\mathrm{Var}(y_t) = (1+\vartheta_1^2)\sigma^2$$

while $\gamma_1 = \vartheta_1 \sigma^2$. The model can also be written as:

$$y_t = \mu + \tilde{\varepsilon}_{t-1} - \frac{1}{\vartheta_1}\tilde{\varepsilon}_t, \qquad \tilde{\varepsilon}_t \sim \mathcal{N}(0, \vartheta_1^2 \sigma^2).$$

Both pairs (ϑ_1, σ) and $(1/\vartheta_1, \vartheta_1 \sigma)$ lead to alternative representations of the same model. This may justify the restriction to $|\vartheta_1| < 1$.

Contrary to the AR(p) model, the MA(q) model is not per se Markovian, although it can be represented in state space form (see below). Although we find that $y_{1:T}$ is a normal random variable with constant mean μ and covariance matrix

$$\Sigma = \begin{pmatrix} \sigma^2 & \gamma_1 & \gamma_2 & \cdots & \gamma_q & 0 & \cdots & 0 & 0 \\ \gamma_1 & \sigma^2 & \gamma_1 & \cdots & \gamma_{q-1} & \gamma_q & \cdots & 0 & 0 \\ & & & \ddots & & & & & \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & \gamma_1 & \sigma^2 \end{pmatrix},$$

where

$$\gamma_s = \sigma^2 \sum_{i=0}^{q-|s|} \vartheta_i \vartheta_{i+|s|}$$

for $|s| \leq q$, and thus provides an explicit likelihood function, this is not manageable in practice. The computation and integration (or maximization) of the likelihood is quite costly because it involves inverting the $n \times n$-matrix Σ. A more manageable representation is to use the likelihood of $y_{1:T}$ conditional on $(\varepsilon_0, \ldots, \varepsilon_{-q+1})$,

$$\ell(\mu, \vartheta_1, \ldots, \vartheta_q, \sigma \,|\, y_{1:T}, \varepsilon_0, \ldots, \varepsilon_{-q+1}) = \\ \sigma^{-T} \prod_{t=1}^{T} \exp\left[-\left(y_t - \mu + \sum_{j=1}^{q} \vartheta_j \hat{\varepsilon}_{t-j} \right)^2 / 2\sigma^2 \right],$$

where

$$\hat{\varepsilon}_t = y_t - \mu + \sum_{j=1}^{q} \vartheta_j \hat{\varepsilon}_{t-j}, \quad \hat{\varepsilon}_0 = \varepsilon_0, \ldots, \hat{\varepsilon}_{1-q} = \varepsilon_{1-q}$$

for $t > 0$. This recursive definition of the likelihood is still costly with $O(Tq)$. Nevertheless, using simualtion methods (like MCMC) is much more feasable than with the first representation.

Another approach is the state-space representation

$$\begin{aligned} x_{t+1} &= F_t x_t + u_t, && \text{(State equation)}, \\ y_t &= G_t x_t + v_t, && \text{(Observation equation)}, \end{aligned}$$

where u_t and v_t are multivariate normal vectors with zero mean and general covariance matrices that may depend on t (such that $\mathbf{E}[u_t v_\tau^\top] = 0$ for all t and τ). It is in the case of the MA(q) model given by:

$$x_t = (\varepsilon_{t-q}, \ldots, \varepsilon_{t-1}, \varepsilon_t)^\top$$

and

$$x_{t+1} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ & & & \cdots & \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} x_t + \varepsilon_{t+1} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix},$$

$$y_t = \mu - \begin{pmatrix} \vartheta_q & \vartheta_{q-1} & \cdots & \vartheta_1 & -1 \end{pmatrix} x_t.$$

We will study general state space models in full detail in section 3.6. With the given representation, the recursive methods provided there can be used. It should be noted that the state space representation of a model is not unique.

Whatever the representation chosen for the MA(q) model, if we want the identifiability condition on $Q(y)$ to hold, then the ϑ_j's vary in a complicated space which cannot be described for values q larger than 3. The reparemeterization given by the Durbin-Levinson recursion also formally applies to this case, with a different interpretation of the ψ_i's which are then the ***inverse partial autocorrelations*** (Jones [1987]). A uniform prior on the ψ_i's can be used for the estimation of the ϑ_i's; then necessarily simulation methods (like MCMC) have to be used.

The ARMA(p,q) model A straightforward extension of the previous model is given by the ***autoregressive moving average model*** ARMA(p,q):

$$y_t = \mu - \sum_{i=1}^{p} \rho_i (y_{t-i} - \mu) + \varepsilon_t - \sum_{j=1}^{q} \vartheta_j \varepsilon_{t-j}, \qquad \varepsilon_t \sim \mathcal{N}(0, \sigma^2).$$

These models, compared to AR and MA models, are aimed at parsimonity, i.e. to use much smaller values of p or q than in the pure AR or MA models.

Stationarity and identifiability constraints still correspond to the polynomials P and Q to be outside the unit circle, with the further condition that both polynomials have no common root. But this last event almost surely never happens under a continuous prior on the parameters. The reparameterization according to the Durbin-Levinson recursion applies therefore for both the ρ_i's and the ϑ_i's (and then using MCMC methods for simulation).

A possible state space representation for the ARMA(p,q) model is given by

$$x_{t+1} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ & & & \cdots & \\ 0 & 0 & 0 & \cdots & 1 \\ \rho_r & \rho_{r-1} & \rho_{r-2} & \cdots & \rho_1 \end{pmatrix} x_t + \varepsilon_{t+1} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix},$$

$$y_t = \mu - \begin{pmatrix} \vartheta_{r-1} & \vartheta_{r-2} & \cdots & \vartheta_1 & -1 \end{pmatrix} x_t$$

with $r := \max(p, q+1)$ and the convention that $\rho_t = 0$ if $t > p$ and $\vartheta_t = 0$ if $t > q$. This representation is again applicable to the methods presented in section 3.6.

3.4.3 Markov chains

Markov processes serve as backbones for both stochastic models through time and as tools for computations in Markov Chain Monte Carlo (MCMC) methods. We follow Meyn and Tweedie [1993] and Cappé et al. [2005].

Some notions A Markov chain $X = \{X_0, X_1, \dots\}$ is a particular type of stochastic process taking, at times $n \in \mathbb{N}$, values X_n in a ***state space*** $\mathscr{X}$.

A discrete time stochastic process X on a state space is a collection $X = (X_0, X_1, \dots)$ of random variables, with each X_n taking values in $\mathscr{X}$; these random variables are assumed to be measurable individually with respect to some given σ-algebra $\mathfrak{B}(\mathscr{X})$.

When thinking of the process as an entity, we regard values of the whole chain X itself, called ***sample paths*** or ***realizations***, as lying in the ***sequence space*** or ***path space*** formed by a countable product $\Omega = \mathscr{X}^\infty = \prod_{n=0}^{\infty} \mathscr{X}_n$, where each $\mathscr{X}_n$ is a copy of $\mathscr{X}$ equipped with a copy of $\mathfrak{B}(\mathscr{X})$. For X to be defined as a random variable in its own right, Ω will be equipped with a σ-algebra $\mathfrak{F}$, and for each state $x \in \mathscr{X}$ thought of as an initial condition in the sample path, there will be a probability measure P_x such that the probability of the event $\{X \in A\}$ is well-defined for any set $A \in \mathfrak{F}$; the initial condition requires, of course, that $\mathsf{P}_x(X_0 = x) = 1$.

The triple $\{\Omega, \mathfrak{F}, \mathsf{P}_x\}$ thus defines a stochastic process since $\Omega = \{\omega_0, \omega_1, \dots \mid \omega_i \in \mathscr{X}\}$ has the product structure to enable the projections ω_n at time n to be well defined realizations of the random variables X_n.

State space definitions

(1) The state space $\mathscr{X}$ is called ***countable*** if $\mathscr{X}$ is discrete, with a finite or countable number of elements, and with $\mathfrak{B}(\mathscr{X})$ the σ-algebra of all subsets of $\mathscr{X}$.

(2) The state space $\mathscr{X}$ is called ***general*** if it is equipped with a countably generated σ-algebra $\mathfrak{B}(\mathscr{X})$.

(3) The state space $\mathscr{X}$ is called ***topological*** if it is equipped with a locally compact, separable, metrizable topology with $\mathfrak{B}(\mathscr{X})$ as the Borel σ-algebra.

Usually results can be obtained for general state spaces and are afterwards applied to the (more structured) topological state spaces (therefore the order of the enumeration). Topological state spaces are the state spaces encountered most often in the applications.

Markov chains in general state spaces Let $\mathscr{X}$ be a general set, and $\mathfrak{B}(\mathscr{X})$ denote a countably generated σ-algebra on $\mathscr{X}$: when $\mathscr{X}$ is topological, then $\mathfrak{B}(\mathscr{X})$ will be taken as the Borel σ-algebra, but otherwise it may be arbitrary. We can then define (Meyn and Tweedie [1993]):

Definition 3.11 (Transition probability kernels)**:** *If* $P = \{P(x,A) \mid x \in \mathscr{X}, A \in \mathfrak{B}(\mathscr{X})\}$ *such that*

(i) *for each* $A \in \mathfrak{B}(\mathscr{X})$, $P(\cdot,A)$ *is a non-negative measurable function on* $\mathscr{X}$*, and*

(ii) *for each* $x \in \mathscr{X}$, $P(x,\cdot)$ *is a probability measure on* $\mathfrak{B}(\mathscr{X})$*,*

then we call P *a* ***transition probability kernel*** *or* ***Markov transition function****.*

We first define a finite sequence $X = (X_0, X_1, \ldots, X_n)$ of random variables on the product space $\mathscr{X}^{n+1} = \prod_{i=0}^{n} \mathscr{X}_i$, equipped with the product σ-algebra $\bigvee_{i=0}^{n} \mathfrak{B}(\mathscr{X}_i)$, by an inductive procedure.

For any measurable sets $A_i \subseteq \mathscr{X}_i$, we develop the set functions $\mathsf{P}_x^n(\cdot)$ on $\mathscr{X}^{n+1}$ by setting, for a fixed starting point $x \in \mathscr{X}$ and for the "cylinder sets" $A_1 \times \cdots \times A_n$

$$\begin{aligned}
\mathsf{P}_x^1(A_1) &= P(x,A_1),\\
\mathsf{P}_x^2(A_1 \times A_2) &= \int_{A_1} P(x,dy_1)P(y_1,A_2),\\
&\vdots\\
\mathsf{P}_x^n(A_1 \times \cdots \times A_n) &= \int_{A_1} P(x,dy_1) \int_{A_2} P(x,dy_2) \cdots P(y_{n-1},A_n).
\end{aligned}$$

These are well-defined by the measurability of the integrands $P(\cdot,\cdot)$ in the first variable, and the fact that the kernels are measures in the second variable.

If we now extend P_x^n to all of $\bigvee_{i=0}^{n} \mathfrak{B}(\mathscr{X}_i)$ in the usual way and repeat this procedure for increasing n, we find (Meyn and Tweedie [1993]):

Theorem 3.9: *For any initial measure* μ *on* $\mathfrak{B}(\mathscr{X})$*, and any transition probability kernel* $P = \{P(x,A) \mid x \in \mathscr{X}, A \in \mathfrak{B}(\mathscr{X})\}$*, there exists a stochastic process* $X = (X_0, X_1, \ldots)$ *on* $\Omega = \prod_{i=0}^{\infty} \mathscr{X}_i$*, measurable with respect to* $\mathfrak{F} = \bigvee_{i=0}^{\infty} \mathfrak{B}(\mathscr{X}_i)$*, and a probability measure* P_μ *on* $\mathfrak{F}$ *such that* $\mathsf{P}_\mu(B)$ *is the probability of the event* $\{X \in B\}$ *for* $B \in \mathfrak{F}$*; and for measurable* $A_i \subseteq \mathscr{X}_i$*,* $i = 0, \ldots, n$*, and any* n

$$\begin{aligned}
&\mathsf{P}_\mu(X_0 \in A_0, X_1 \in A_1, \ldots, X_n \in A_n)\\
&\qquad = \int_{y_0 \in A_0} \cdots \int_{y_{n-1} \in A_{n-1}} \mu(dy_0)P(y_0,dy_1) \cdots P(y_{n-1},A_n).
\end{aligned}$$

This last equation will be the defining property of Markov chains (Meyn and Tweedie [1993]):

Definition 3.12 (Markov chains in general state spaces)**:** *The stochastic process* X *defined on* $(\Omega,\mathfrak{F})$ *is called a* ***time-homogeneous Markov chain*** *with* ***transition probability kernel*** $P(x,A)$ *and* ***initial distribution*** μ *if the finite-dimensional distributions of* X *satisfy the equation in the foregoing theorem for every* n*.*

The n-step transition probability kernel The ***n-step transition probability kernel*** is defined iteratively. We set $P^0(x,A) = \delta_x(A)$, the Dirac measure defined by

$$\delta_x(A) = \begin{cases} 1 & x \in A, \\ 0 & x \notin A \end{cases}$$

and, for $n \geq 1$, we define inductively

$$P^n(x,A) = \int_{\mathscr{X}} P(x,dy)P^{n-1}(y,A), \qquad \text{for } x \in \mathscr{X}, A \in \mathfrak{B}(\mathscr{X}).$$

We write P^n for the n-step transition probability kernel $\{P^n(x,A) \mid x \in \mathscr{X}, A \in \mathfrak{B}(\mathscr{X})\}$.

As an application of the construction equations, we have the celebrated Chapman-Kolmogorov equations, which are fundamental for the fact that many results can be transferred from the countable to the general case.

Theorem 3.10: *For any m with $0 \leq m \leq n$,*

$$P^n(x,A) = \int_{\mathscr{X}} P^m(x,dy)P^{n-m}(y,A), \qquad \textit{for } x \in \mathscr{X}, A \in \mathfrak{B}(\mathscr{X}).$$

We can alternatively write this as

$$\mathsf{P}_x(X_n \in A) = \int_{\mathscr{X}} \mathsf{P}_x(X_m \in dy)\mathsf{P}_y(X_{n-m} \in A).$$

Exactly as the one-step transition probability kernel describes a chain X, the m-step kernel (viewed in isolation) satisfies the definition of a transition kernel, and thus defines a Markov chain $X^m = \{X_n^m\}$ with transition probabilities

$$\mathsf{P}_x(X_n^m \in A) = P^{mn}(x,A).$$

Stability analysis of Markov chains

Meyn and Tweedie [1993] use the term "stability" in connection with Markov chains (or more generally with stochastic processes) as a basic concept that serves to cover a wide range of similar but not identical ideas of "stable" behaviour (in an intuitive sense) of the considered processes. The stability concepts are also related to similar considerations in dynamical or stochastic systems theory, which is concerned with the same questions but under different assumptions on the model structures.

Stopping times The behaviour of a Markov chain involves the distributions at certain random times in its evolution, generally called ***stopping times***. Particular instances of stopping times are (Meyn and Tweedie [1993]):

Definition 3.13 (Hitting times, return times, occupation times)**:** ***(i)*** *For any set $A \in \mathfrak{B}(\mathscr{X})$, we call*

$$\sigma_A := \min\{n \geq 0 \mid X_n \in A\}$$
$$\tau_A := \min\{n \geq 1 \mid X_n \in A\}$$

*the **first hitting time** and **first return** on A, respectively. Here, we set* $\min\emptyset = +\infty$ *by convention.*

(ii) *For any set* $A \in \mathfrak{B}(\mathscr{X})$*, the **occupation time*** η_A *is the number of visits by* X *to* A *after time zero, given by*

$$\eta_A := \sum_{n=1}^{\infty} \mathbf{1}_A(X_n).$$

For every $A \in \mathfrak{B}(\mathscr{X})$, σ_A, τ_A and η_A are measurable functions from Ω to $\mathbb{N} \cup \{\infty\}$.
The stability analysis on X involves the following kernels for $x \in \mathscr{X}$ and $A \in \mathfrak{B}(\mathscr{X})$:

- We define U as the expected value of the occupation times,

$$U(x,A) := \sum_{n=1}^{\infty} P^n(x,A) = \mathbf{E}_x[\eta_A]$$

  which maps $\mathscr{X} \times \mathfrak{B}(\mathscr{X})$ to $\mathbb{R} \cup \{\infty\}$.

- We define L as the return time probabilities

$$L(x,A) := \mathsf{P}_x(\tau_A < \infty) = \mathsf{P}_x(X \text{ ever enters } A).$$

- We consider the event that $X \in A$ infinitely often (i.o.), or $\eta_A = \infty$, defined by

$$\{X \in A \text{ i.o.}\} := \bigcap_{N=1}^{\infty} \bigcup_{k=N}^{\infty} \{X_k \in A\}$$

  which is well-defined as an $\mathfrak{F}$-measurable event on Ω. Then we define Q as

$$Q(x,A) := \mathsf{P}_x\{X \in A \text{ i.o.}\}.$$

Obviously, for any $x \in \mathscr{X}$, $A \in \mathfrak{B}(\mathscr{X})$, we have $Q(x,A) \le L(x,A)$. Nevertheless, it can be shown (Meyn and Tweedie [1993], Prop. 9.1.1) that if for any set $A \in \mathfrak{B}(\mathscr{X})$ we have $L(x,A) = 1$ for all $x \in \mathscr{X}$, then we have $Q(x,A) = L(x,A) = 1$ for all $x \in \mathscr{X}$. We thus have an equivalence

$$\mathsf{P}_x(\tau_A < \infty) = 1 \text{ for all } x \in \mathscr{X} \quad \Leftrightarrow \quad \mathsf{P}_x(\eta_A = \infty) = 1 \text{ for all } x \in \mathscr{X}.$$

Stochastic stability for Markov chains Meyn and Tweedie [1993] refer to the following concepts of "stability" for a general state space Markov chain X:

(I) ***φ-irreducibility***: There exists a measure φ on $\mathfrak{B}(\mathscr{X})$ with the property that for every starting point $x \in \mathscr{X}$

$$\varphi(A) > 0 \quad \Rightarrow \quad \mathsf{P}_x(\tau_A < \infty) > 0.$$

This condition ensures that all "reasonable sized" sets (measured by φ), can be reached from every possible starting point.

For a countable space chain, φ-irreducibility is just the concept of irreducibility commonly used with φ taken as the counting measure.

For a state space model, φ-irreducibility is related to the idea that we are able to "steer" the system to every other state in $\mathbb{R}^n$ (for deterministic linear control systems, this property is related to controllability). If this does not hold then for some starting points one gets stuck in one part of the space forever; other starting points lead to disjoint regions of the space where one stays forever. Irreducibility precludes this.

Thus irreducibility ensures a stability in the sense that if we have a small change in the starting point, the system does not suddenly change to a completely different and disjoint set of states which are not reachable from each other.

(II) ***Recurrence***: There exists a measure φ such that for every starting point $x \in \mathscr{X}$

$$\varphi(A) > 0 \quad \Rightarrow \quad \mathbf{E}_x[\eta_A] = \infty.$$

This weakest form of recurrence is based on the occupation time η_A which counts the number of visits to a set A which in the case of a recurrent chain happens in expectation infinitely often.

Some stronger conditions for recurrence are possible: the Harris recurrence

$$\varphi(A) > 0 \quad \Rightarrow \quad \mathsf{P}_x(\tau_A < \infty) = 1,$$

which is equivalent to

$$\varphi(A) > 0 \quad \Rightarrow \quad \mathsf{P}_x(\eta_A = \infty) = 1,$$

or, even stronger, that for every starting point $x \in \mathscr{X}$

$$\varphi(A) > 0 \quad \Rightarrow \quad \mathbf{E}_x[\tau_A] < \infty.$$

These conditions ensure that reasonable sized sets are reached with probability one (first possibility), or even in finite mean time (second possibility). Thus, these requirements guarantee not only the possibility of reaching states, but that reaching such sets of states is guaranteed eventually. For deterministic models, the last two recurrence concepts are the same, for stochastic models they are definitely different (***evanescence*** in the first case and ***tightness*** in the second). All conditions have the heuristic interpretation that the chain returns to the "center" of the space in a recurring way (in the last case only faster than in the other cases). In all cases, the chain does not just drift off (or evanesce) away from the center of the state space.

(III) ***Ergodicity***: this is the limiting behaviour of the chain. It emerges that in the stronger recurrent situation, there is an "invariant regime" described by a measure p such that if the chain starts in this regime (that is, if X_0 has distribution p) then it remains in the regime, and moreover if the chain starts in some other regime that it converges in a strong probabilistic sense with p as a limiting distribution.

In the following, we will give further discussions of the concepts of irreducibility, recurrence, and ergodicity, eventually leading to the important ergodicity theorem.

Irreducibility

The ***irreducibility*** concept for Markov chains with *countable* state spaces requires that for any point $x \in \mathscr{X}$ the probability of reaching any other point $y \in \mathscr{X}$ is positive:

$$\mathsf{P}_x(\sigma_y < \infty) > 0.$$

This concept cannot directly be adopted to general state space Markov chains, because the probability of reaching a single point y in the state space is typically zero.

φ-irreducibility

We follow again Meyn and Tweedie [1993] and Cappé et al. [2005].

Definition 3.14 (φ-irreducibility)**:** *We call $X = \{X_n\}$ φ-**irreducible** if there exists a measure φ on $\mathfrak{B}(\mathscr{X})$ such that for all $x \in \mathscr{X}$*

$$\varphi(A) > 0 \quad \Rightarrow \quad L(x,A) > 0.$$

*We call such a measure an **irreducibility measure**.*

For countable state spaces, this condition is weaker than irreducibility. There are a number of alternative formulations of φ-irreducibility. Define the transition kernel

$$K_{a_{\frac{1}{2}}}(x,A) := \sum_{n=0}^{\infty} P^n(x,A) 2^{-(n+1)}, \qquad x \in \mathscr{X}, A \in \mathfrak{B}(\mathscr{X}).$$

This kernel $K_{a_{\frac{1}{2}}}$ defines for each x a probability measure equivalent to $\sum_{n=0}^{\infty} P^n(x,A)$, which may be infinite for many sets A.

Theorem 3.11: *The following are equivalent formulations of φ-irreducibility:*

- ***(i)*** *for all $x \in \mathscr{X}$, whenever $\varphi(A) > 0$, then $U(x,A) > 0$;*
- ***(ii)*** *for all $x \in \mathscr{X}$, whenever $\varphi(A) > 0$, there exists some $n > 0$, possibly depending on both A and x, such that $P^n(x,A) > 0$;*
- ***(iii)*** *for all $x \in \mathscr{X}$, whenever $\varphi(A) > 0$, then $K_{a_{\frac{1}{2}}}(x,A) > 0$.*

Maximal irreducibility measure

Theorem 3.12: *If X is φ-irreducible for some measure φ, then there exists a probability measure ψ on $\mathfrak{B}(\mathscr{X})$ such that*

- ***(i)*** *X is ψ-irreducible;*
- ***(ii)*** *for any other measure φ', the chain X is φ'-irreducible if and only if $\psi \succ \varphi'$;*
- ***(iii)*** *if $\psi(A) = 0$, then $\psi(\{y \mid L(y,A) > 0\}) = 0$;*

(iv) *the probability measure ψ is equivalent to*

$$\psi'(A) := \int_{\mathscr{X}} \varphi'(dy) K_{a_{\frac{1}{2}}}(y, A)$$

for any finite irreducibility measure φ'.

Such a ψ is called ***maximal irreducibility measure***. We write

$$\mathfrak{B}^+(\mathscr{X}) := \left\{ A \in \mathfrak{B}(\mathscr{X}) \;\middle|\; \psi(A) > 0 \right\}$$

for the sets of positive ψ-measure, called ***accessible sets***; the equivalence of maximal irreducibility measures means that $\mathfrak{B}^+(\mathscr{X})$ is uniquely defined.

Recurrence

We follow Meyn and Tweedie [1993] and Cappé et al. [2005].

Uniform transience and recurrence

Definition 3.15 (Uniform transience and recurrence)**:** *A set $A \in \mathfrak{B}(\mathscr{X})$ is called **uniformly transient** if*

$$\sup_{x \in A} \mathbf{E}_x[\eta_A] < \infty.$$

*A set $A \in \mathfrak{B}(\mathscr{X})$ is called **recurrent** if*

$$\mathbf{E}_x[\eta_A] = \infty \qquad \text{for all } x \in A.$$

An alternative equivalent definition for a uniformly transient set is given by

$$\sup_{x \in \mathscr{X}} \mathbf{E}_x[\eta_A] < \infty,$$

where the supremum is over all $x \in \mathscr{X}$. The main result on φ-irreducible transition kernels is the following recurrence/transience dichotomy (see e.g. Meyn and Tweedie [1993]):

Theorem 3.13: *Let X be a φ-irreducible Markov chain. Then either of the following two statements holds true:*

(i) *Every set in $\mathfrak{B}^+(\mathscr{X})$ is recurrent, in which case we call X **recurrent**.*

(ii) *There is a countable cover of $\mathscr{X}$ with uniformly transient sets, in which case we call X **transient**.*

Invariant measures and stationarity It is clear that in general a Markov chain is not stationary. Nevertheless, it is possible that with an appropriate choice of the initial distribution for X_0 we may produce a stationary process $(X_n)_{n\in\mathbb{N}}$. From the Markov property, it follows that the chain X is stationary if and only if the distribution of X_n does not vary with time n. Such considerations lead to invariant measures.

Definition 3.16 (Invariant measures)**:** *A σ-finite measure p on $\mathfrak{B}(\mathscr{X})$ with the property*

$$p(A) = \int_{\mathscr{X}} p(dx)P(x,A), \qquad A \in \mathfrak{B}(\mathscr{X})$$

*will be called **invariant**.*

Given an initial invariant distribution p, we get

$$p(A) = \int_{\mathscr{X}} p(dx)P^n(x,A) = \mathsf{P}_p(X_n \in A)$$

for any n and all $A \in \mathfrak{B}(\mathscr{X})$.

There is the following existence theorem (Meyn and Tweedie [1993] Th. 10.0.1, Cappé et al. [2005], Th. 14.2.25):

Theorem 3.14: *If the φ-irreducible chain X is recurrent then it admits a unique (up to a multiplicative constant) invariant measure p, and the measure p has the representation, for any $B \in \mathfrak{B}^+(\mathscr{X})$*

$$p(A) = \int_B p(dy)\mathbf{E}_y\left[\sum_{n=1}^{\tau_B} \mathbf{1}_A(X_n)\right], \qquad A \in \mathfrak{B}(\mathscr{X}).$$

This measure is also a maximal irreducibility measure.

If an invariant measure is finite (rather than merely σ-finite), then it may be normalized to a stationary probability measure, and in practice this is the main stable situation of interest. If an invariant measure has infinite total mass, then its probabilistic interpretation is much more difficult, although for recurrent chains, there is at least the interpretation as described in the above theorem.

It is of course not yet clear that an invariant *probability* measure p ever exists, or whether it will be unique when it does exist.

Invariant probability measures are important not merely because they define stationary processes. They will also turn out to be the measures which define the long-term or ergodic behaviour of the chain.

These results lead to define the following classes of chains:

Definition 3.17 (Positive and null chains)**:** *Suppose that X is φ-irreducible, and admits an invariant probability measure p. Then X is called a **positive** chain.*

*If X does not admit such a measure, then we call X **null**.*

If the chain X is positive then it is recurrent. Therefore positive chains are also called ***positive recurrent***.

Harris recurrence It is sometimes useful to consider stronger recurrence properties.

Definition 3.18 (Harris recurrence)**:** *The set A is called* ***Harris recurrent*** *if*

$$Q(x,A) = \mathsf{P}_x(\eta_A = \infty) = 1, \qquad x \in A.$$

A chain X is called ***Harris (recurrent)*** *if it is φ-irreducible and every set in* $\mathfrak{B}^+(\mathscr{X})$ *is Harris recurrent.*

An alternative equivalent definition of Harris recurrence is to replace the condition $Q(x,a) = 1$ by

$$L(x,A) = \mathsf{P}_x(\tau_A < \infty) = 1, \qquad x \in A$$

(see the beginning of this section). A Harris recurrent chain is recurrent. The converse is not true, but one has the following decomposition theorem for recurrent Markov chains (Meyn and Tweedie [1993]), Th. 9.1.5; see also Cappé et al. [2005], Th. 14.2.23):

Theorem 3.15: *Let X be a recurrent Markov chain on a general state space* $\mathscr{X}$ *and let ψ be a maximal irreducibility measure. Then*

$$\mathscr{X} = H \dot{\cup} N$$

where every subset $A \in H \cap \mathfrak{B}^+(\mathscr{X})$ *is Harris recurrent and N is covered by a countable family of uniformly transient sets and* $\psi(N) = 0$.

Definition 3.19 (Positive Harris chains)**:** *If X is Harris recurrent and positive, then X is called a* ***positive Harris*** *chain.*

Ergodicity

We follow again Meyn and Tweedie [1993] and Cappé et al. [2005].

Modes of convergence We consider convergence of a chain in terms of its transition probabilities, although it is important also to consider convergence of a chain along its sample paths, leading to strong laws, or of normalized variables leading to central limit theorems and associated results. This is in contrast to the traditional approach in the countable state space case. Typically, there, the search is for conditions under which there exist pointwise limits of the form

$$\lim_{n\to\infty} |P^n(x,y) - p(y)| = 0;$$

but the results we state in the next paragraphs are related to the signed measure $(P^n - p)$, and so concern not merely such pointwise or even setwise convergence, but a more global convergence in terms of the total variation norm.

Definition 3.20 (Total Variation Norm)**:** *If* μ *is a signed measure on* $\mathfrak{B}(\mathscr{X})$ *then the* ***total variation norm*** $\|\mu\|_{TV}$ *is defined as*

$$\|\mu\|_{TV} := \sup_{f:|f|\le 1} |\mu(f)| = \sup_{A\in\mathfrak{B}(\mathscr{X})} \mu(A) - \inf_{A\in\mathfrak{B}(\mathscr{X})} \mu(A).$$

The key limit of interest to us will be of the form

$$\lim_{n\to\infty} \|P^n(x,\cdot) - p\|_{\mathrm{TV}} = 2 \lim_{n\to\infty} \sup_{A\in\mathfrak{B}(\mathscr{X})} |P^n(x,A) - p(A)| = 0.$$

Obviously when the $\|\cdot\|_{\mathrm{TV}}$-convergence holds on a countable space, then the $|\cdot|$-convergence also holds and indeed holds uniformly in the end-point y. This move to the total variation norm, necessitated by the typical lack of structure of pointwise transitions in the general state space, actually proves exceedingly fruitful rather than restrictive.

When the space is topological, it is also the case that total variation convergence implies weak convergence of the measures in question. This is clear since the latter is defined as convergence of expectations of functions which are not only bounded but also continuous.

Periodicity Let X be a φ-irreducible Markov chain on $\mathscr{X}$ and let ψ be a maximal irreducibility measure. Sets $D_0,\ldots,D_{d-1} \in \mathfrak{B}(\mathscr{X})$ are called to be a ***d-cycle*** of X if

(i) for $x \in D_i$, $P(x, D_{i+1 \bmod d}) = 1, \quad i = 0,\ldots,d-1$;

(ii) the set $N = \complement[\bigcup_{i=0}^{d-1} D_i]$ is ψ-null.

Definition 3.21 (Periodic and aperiodic chains)**:** *Suppose that X is a φ-irreducible Markov chain. The largest d for which a d-cycle occurs for X is called the **period** of X.*

*When $d = 1$, the chain X is called **aperiodic**.*

One can concentrate almost exclusively on aperiodic chains, since:

Theorem 3.16: *Suppose X is a φ-irreducible chain with maximal irreducibility measure ψ, period d and d-cycle $\{D_i, i = 0,\ldots,d-1\}$. Then each of the sets D_i is an absorbing ψ-irreducible set for the chain X_d corresponding to the transition probability kernel P^d, and X_d on each D_i is aperiodic.*

The ergodic theorem We state now the main result, see e.g. Meyn and Tweedie [1993], Th. 13.3.3, or Cappé et al. [2005] Th. 14.2.37:

Theorem 3.17 (Ergodicity Theorem)**:** *Let P be a φ-irreducible positive aperiodic transition kernel with invariant probability measure p. Then for p-almost all x*

$$\lim_{n\to\infty} \|P^n(x,\cdot) - p\|_{TV} = 0.$$

If P is Harris recurrent, the convergence occurs for all $x \in \mathscr{X}$.

Proofs of the theorem can be found in Athreya et al. [1996] as well as in Rosenthal [1995] based on pathwise coupling (Rosenthal [2001] and Roberts and Rosenthal [2004]; see also Cappé et al. [2005]). This result does not provide any information on the rate of convergence. But the conditions are quite minimal, and in fact essentially necessary and sufficient: If $\|P^n(x,\cdot) - p\|_{\mathrm{TV}} \longrightarrow 0$ for every $x \in \mathscr{X}$, then the chain is p-irreducible, aperiodic, positive Harris, and p is an invariant distribution (Nummelin [1984]).

The ergodicity theorem is of particular interest in cases where the invariant distribution is explicitly known. This is the case with Markov chain Monte Carlo algorithms, which will be treated in a subsequent section. The theorem provides conditions that are simple and easy to verify, and under which an MCMC algorithm converges to its stationary distribution.

3.4.4 Graphical models

Graphical models are complex hierarchical models where dependencies of occurring variables are represented by a graph. The combination with graph theory provides a powerful language for specifying and understanding complex statistical models. We follow mainly the short introduction of Green [2001]; for a longer introduction see e.g. Lauritzen [2001].

A ***graphical model*** is a graph (V,E), where the vertices represent (are) variables and the edges (directed or not) represent conditional dependence properties. We will normally use V as an index set and denote the variables by x_v, $v \in V$, but of course index set and variables are one-to-one and thus exchangeable from a mathematical viewpoint.

Directed acyclic graphs Let us first consider the case where the graph is a directed acyclic graph (DAG), i.e. all edges are directed and there are no (directed) cycles (see section 1.4.1). Directed acyclic graphs are a natural representation of the way we usually specify a statistical model, where directions are naturally given by time or cause-and-effect chains (past $\rightarrow$ future, disease $\rightarrow$ symptom, parameters $\rightarrow$ data). A DAG thus expresses the natural factorization of a joint distribution of a variable x_v given the values of its parents $x_{\mathrm{pa}(v)}$:

$$\pi(x) = \prod_{v \in V} \pi(x_v \,|\, x_{\mathrm{pa}(v)}).$$

This in turn implies a Markov property: variables are conditionally independent of their non-descendants given their parents.

A major rôle in MCMC methods is played by the ***full conditionals*** $\pi(x_v \,|\, x_{-v})$ for $v \in V$ where

$$x_{-v} := \left\{ x_u \;\middle|\; u \in V \setminus \{v\} \right\}.$$

Graphical models help in identifying which terms need be included in a full conditional:

$$\pi(x_v \,|\, x_{-v}) \propto \pi(x_v \,|\, x_{\mathrm{pa}(v)}) \prod_{w \text{ with } v \in \mathrm{pa}(w)} \pi(x_w \,|\, x_{\mathrm{pa}(w)}).$$

This involves one term for the variable of interest itself and one term for each of its children.

Undirected graph, and spatial modelling Sometimes it is necessary to give up the directed dependencies between variables, because (Green [2001])

- there is no natural direction (e.g. in spatial models),
- in understanding associations between variables, directions can confuse, and
- these *associations* represent the full conditionals needed in setting up MCMC methods.

To form a conditional independence graph for a multivariate distribution, draw an (undirected) edge between variables α and β if they are not conditionally independent given all other variables.

Markov properties For a temporal stochastic process there are many equivalent ways to express the Markov property. A stochastic process can be seen as a simple graphical model. For general graphical models, the situation is more complicated: one can distinguish four different related properties, which show to be equivalent in the case of temporal stochastic processes (see Lauritzen [2001]):

- P: ***Pairwise Markov property*** Non-adjacent pairs of variables are conditionally independent given the rest (this is how the graph is made up);
- L: ***Local Markov property*** Conditionally on adjacent variables (neighbours), each variable is independent of all others (so that full conditionals are simplified);
- G: ***Global Markov property*** Any two subsets of variables separated by a third are conditionally independent given the values of the third subset;
- F: ***Factorization*** The joint distribution factorizes as a product of functions on cliques (i.e. maximal complete subgraphs, see section 1.4.1).

It is always true that

$$F \Rightarrow G \Rightarrow L \Rightarrow P,$$

but these four Markov properties are in general different. However, in many statistical contexts, the four properties are the same. A sufficient but not necessary condition is that the joint distribution has the positivity property ("any values realizable individually are realizable jointly"). This results from the Clifford-Hammersley theorem (Markov random field = Gibbs distribution, $L = F$). (A typical context in which Markov properties do not coincide is where there are logical implications between some subsets of variables.)

For DAGs, we have always at least $L = F$ if a reference product measure exists.

Modelling with an undirected graph With a DAG, because of acyclicity, any set of conditional distributions $\pi(x_v \,|\, x_{\mathrm{pa}(v)})$ combine to form a consistent joint distribution. In an undirected graph, one needs consistency conditions on the full conditionals $\pi(x_v \,|\, x_{-v})$ (using L, this is equal to $\pi(x_v \,|\, x_{\mathrm{ne}(v)})$ where $\mathrm{ne}(v)$ denotes the neighbours of v). The only safe strategy is to use property F, to model the joint distribution as a product of functions on cliques

$$\pi(x) = \prod_C \psi_C(x_C).$$

We can then use property L to read off the full conditionals needed to set up MCMC:

$$\pi(x_v \,|\, x_{-v}) = \prod_{C \text{ with } v \in C} \psi_C(x_C) = \pi(x_v \,|\, x_{\mathrm{ne}(v)}).$$

Chain graphs In hierarchical spatial models, one needs a hybrid modelling strategy: there will be some directed and some undirected edges. If there are no one-way cycles, the graph can be arranged to form a DAG with composite nodes called ***chain components*** Δ_t that are connected subgraphs remaining when all directed edges are removed: we call this a ***chain graph***.

Model specification uses a combination of approaches using DAGs and undirected graphs. This builds a joint distribution

$$\pi(x) = \prod_t \pi(x_{\Delta_t} \,|\, x_{\mathrm{pa}(\Delta_t)}) = \prod_t \prod_{C \in \mathscr{C}_t} \psi_C(x_C)$$

where $\mathscr{C}_t$ are the cliques in an undirected graph with vertices $(\Delta_t, \mathrm{pa}(\Delta_t))$ and undirected edges consisting of

(a) those already in Δ_t,

(b) the links between Δ_t and its parents, with directions dropped, and

(c) links between all members of $\mathrm{pa}(\Delta_t)$.

3.5 Computational issues

Bayesian analysis can be done analytically only in a few cases, for example when conjugate priors are used. In dynamical settings, the situation is even worse. Often only numerical methods are possible: high-dimensional integrals have to be computed. In high dimensions simulations with Monte Carlo methods converge faster than conventional (e.g. quadrature) methods: in the optimal case, the convergence rate does not depend on the dimension. But for these simulations realizations of i.i.d. random variables with complicated distributions have to be produced. This is often very difficult to do directly. The task is easier when the independence is given up: Realizations are given by a Markov Chain. Simple transition distributions yield then complex stationary distributions. But how do we know which transition distributions have to be used? The Gibbs Sampler and Metropolis-Hastings algorithms provide solutions. Nevertheless, for dynamical models, this methods are not appropriate. Here Sequential Monte Carlo (SMC) methods have emerged as a promising tool.

3.5.1 Bayesian calculations

Implementation difficulties occur when (Robert [2001]):

- Computing the posterior distribution

$$\pi(\theta \,|\, y) = \frac{f(y \,|\, \theta)\pi(\theta)}{\int_\Theta f(y \,|\, \theta)\pi(\theta) d\theta}.$$

- Computing the Bayes rule

$$\delta^{\pi}(y) = \arg\min_{d} \mathbf{E}^{\pi}[L(\theta,d)\,|y] = \arg\min_{a} \int_{\Theta} L(\theta,a) f(y\,|\,\theta)\pi(\theta)d\theta.$$

- Maximization of the marginal posterior for MAP estimation

$$\arg\max \int_{\Theta_1\times\cdots\times\widehat{\Theta_i}\times\cdots\times\Theta_d} \pi(\theta\,|y)d\theta_1\cdots\widehat{d\theta_i}\cdots d\theta_d$$

  where $\widehat{\ }$ means "leave out", and we assume $\theta := (\theta_1,\ldots,\theta_d) \in \Theta := \Theta_1\times\cdots\times\Theta_d$.

- Computing posterior quantities

$$\mathbf{E}^{\pi}[g\,|y] = \int_{\Theta} g(\theta)\pi(\theta\,|y)d\theta = \frac{\int_{\Theta} g(\theta) f(y\,|\,\theta)\pi(\theta)d\theta}{\int_{\Theta} f(y\,|\,\theta)\pi(\theta)d\theta}.$$

- Computation of posterior quantiles q of order α, i.e. the solution (in q) of

$$\Pr\left(\pi(\theta\,|y) \geq q\,|y\right) = \alpha.$$

In all cases high-dimensional integrals are involved and are to be solved. Practical computations are not always possible analytically. For conjugate distributions, the posterior expectation of the natural parameters can be expressed analytically, for one or several observations. But even conjugate priors may lead to computational difficulties.

Classical implementations Classical implementations for numerical integration are for example (Robert [2001]):

- Simpson's method;
- Polynomial quadrature;
- Orthogonal Bases;
- Wavelets.

All these approaches bump into the curse of dimensionality if tried to be applied to the computation of the high dimensional integrals of Bayesian analysis.

Monte Carlo methods

The main idea with ***Monte Carlo (MC) methods*** is to approximate a distribution by a possibly weighted mixture of delta distributions. The support of each delta distribution has to be determined through suitable sampling methods. We follow Doucet et al. [2001a].

Empirical distribution The simplest Monte Carlo method is to approximate a target distribution $p(dx)$ with samples drawn from this distribution. Let $x^{(i)}$, $i = 1,\ldots,N$ be independently sampled from $p(dx)$, then the mixture of delta distributions

$$p(dx) \approx \hat{p}_N(dx) := \frac{1}{N}\sum_{i=1}^{N} \delta_{x^{(i)}}(dx), \qquad x_i \sim p(dx) \text{ for } i = 1,\ldots,N,$$

can be seen as an approximation to the target distribution $p(dx)$. One has for example for all suitable functions $h(x)$

$$\mathbf{E}^p[h(x)] = \int h(x)p(dx) \longrightarrow \frac{1}{N}\sum_{i=1}^{N} x^{(i)}$$

for $N \to \infty$, if the variance

$$\sigma_h^2 := \mathrm{Var}_p[h(x)] = \mathbf{E}^p[\big(h(x) - \mathbf{E}^p[h(x)]\big)^2] = \mathbf{E}^p[h^2(x)] - \mathbf{E}^p[h(x)]^2$$

of $h(x)$ with respect to $p(x)$ is finite. It follows that also $\mathrm{Var}_{\hat{p}_N}[h(x)] < \infty$, and with the law of large numbers one gets:

$$\mathbf{E}^{\hat{p}_N}[h(x)] = \frac{1}{N}\sum_{i=1}^{N} \mathbf{E}^p[h(x)] \longrightarrow \mathbf{E}^p[h(x)] \text{ a.s.} \qquad N \longrightarrow \infty.$$

Furthermore, a central limit theorem holds of the form

$$\sqrt{N}\left(\mathbf{E}^{\hat{p}_N}[h(x)] - \mathbf{E}^p[h(x)]\right) \longrightarrow \mathcal{N}(0,\sigma_h^2)$$

(convergence in distribution; cf. Doucet et al. [2001a]). A decisive advantage of this method is that the convergence rate $\sqrt{N}$ appearing in this theorem is independent from the dimension of x. This is in contrast to grid-based methods for the solutions of integrals, thus beating the curse of dimensionality: Grid-based methods converge fairly faster than the MC methods in low dimensions, but the convergence rates decrease rapidly whith increasing dimensions. Monte Carlo methods are thus particularly preferred with high-dimensional problems.

Importance Sampling (IS) For the computation of the empirical distribution of a target distribution $p(dx)$, one has to be able to draw samples from $p(dx)$. Especially in high-dimensional problems, this is seldom possible. In this case, on may use ***Importance Sampling (IS)***: Let $q(dx)$ be an arbitrary distribution, called ***proposal distribution***, with equal or larger support than $p(dx)$, i.e. from $p(dx) \neq 0$ follows also $q(dx) \neq 0$. Then, the ***weighted empirical distribution***

$$\sum_{i=1}^{M} \frac{\tilde{\omega}^{(i)}}{\sum_{j=1}^{M}\tilde{\omega}^{(j)}} \delta_{\tilde{x}^{(i)}}(dx), \qquad \tilde{x}^{(i)} \sim q(dx) \text{ for } i = 1,\ldots,M$$

with M samples $\tilde{x}^{(i)}$ from q and with the ***unnormalized importance weights***

$$\tilde{\omega}^{(i)} := \frac{p(d\tilde{x}^{(i)})}{q(d\tilde{x}^{(i)})}$$

also provide an approximation of $p(dx)$. Here, the quotient $p(dx)/q(dx)$ has to be understood in the Radon-Nikodym sense; if $p(dx)$ and $q(dx)$ are given through probability densities with respect to a σ-finite reference measure μ (we denote the densities also with p and q, respectively), i.e.

$$p(dx) = p(x)d\mu(x), \qquad q(dx) = q(x)d\mu(x),$$

then the importance weights are simply the quotients of the respective densities:

$$\tilde{\omega}^{(i)} = \frac{p(\tilde{x}^{(i)})}{q(\tilde{x}^{(i)})}.$$

The quality (in terms of convergence) of this method is crucially dependend on the proposal distribution $q(dx)$. The choice of q is a critical point of IS.

Sampling/Importance Resampling (SIR) With IS, one draws samples from q, not from p; if one wants to have samples from p, then one can achieve this with an additional ***resampling*** step, also called ***selection***: If samples

$$\tilde{x}^{(1)}, \ldots, \tilde{x}^{(M)}$$

with unnormalized importance weights

$$\tilde{\omega}^{(1)}, \ldots, \tilde{\omega}^{(M)}$$

are given, then, after normalization of the importance weights $\tilde{\omega}^{(i)}$ by

$$\omega^{(i)} := \frac{\tilde{\omega}^{(i)}}{\sum_{j=1}^{M} \tilde{\omega}^{(j)}}, \qquad i = 1, \ldots, M,$$

one draws N independent samples $(x^{(1)}, \ldots, x^{(N)})$ (with replacement) from the M samples $(\tilde{x}_{(i)})$ with probabilities according to the ***normalized importance weights*** $\omega^{(i)}$. Hence, each number N_i counting how often the sample $x^{(i)}$ has been drawn follows a binomial distribution $\mathscr{B}(N, \omega^{(i)})$. The complete vector $(N_1, \ldots, N_M)$ is distributed according to the multinomial distribution $\mathscr{M}(N, \omega^{(1)}, \ldots, \omega^{(M)})$.

Resampling always increases the variance of the empirical estimations and seems thus unnecessary in the given non-sequential context. This is essentially changed for sequential estimations, as it is the case for online estimations of dynamical systems, where the resampling step is of utmost importance. In the next paragraph we will therefore describe some variants of resampling with lower variance. The resulting samples are in all cases approximately i.i.d. $p(dx)$. The algorithm combining importance sampling with a resampling step is called ***Sampling/Importance Resampling (SIR)***, introduced by Rubin [1987].

Alternative and more powerful methods than SIR are the Markov Chain Monte Carlo (MCMC) methods described in the next subsection. But sequential versions of SIR for static estimation being similar to the Sequential Monte Carlo methods for state space systems described in subsection 3.6.4 start to develop to be an important rival to MCMC methods (see Del Moral et al. [2006]).

Alternative selection schemes We use Hol et al. [2006]. It is not necessary to use multinomial resampling in the SIR algorithm. The aim is to replace the weighted empirical density

$$p_M(dx) := \sum_{i=1}^{M} \omega^{(i)} \delta_{\tilde{x}^{(i)}}(dx)$$

with an unweighted empirical density

$$\hat{p}_N(dx) := \frac{1}{N} \sum_{k=1}^{N} \delta_{x^{(k)}}(dx)$$

such that for any suitable function $g(x)$ it holds that

$$\mathbf{E}\left[\left(\mathbf{E}^{p_M}[g] - \mathbf{E}^{\hat{p}_N}[g]\right)^2\right] \longrightarrow 0 \quad \text{if } N, M \to \infty$$

for the expected values

$$\mathbf{E}^{p_M}[g] = \int g(x) p_M(x) dx \qquad \text{and} \qquad \mathbf{E}^{\hat{p}_N}[g] = \int g(x) \hat{p}_N(x) dx,$$

respectively.

Then convergence holds (see e.g. Crisan and Doucet [2002]). This can be achieved with several resampling schemes. Common to them all is the following procedure: One produces the "cumulative" intervals

$$\left[\sum_{s=1}^{i-1} \omega^{(s)}, \sum_{s=1}^{i} \omega^{(s)}\right), \qquad i = 1, \ldots, M$$

with length $\omega^{(i)}$, where we use the convention that empty sums are equal to 0. Then one samples random values u_k for $k = 1, \ldots, N$, and sets

$$x_k := \tilde{x}_i$$

where the index i is determined such that

$$u_k \in \left[\sum_{s=1}^{i-1} \omega^{(s)}, \sum_{s=1}^{i} \omega^{(s)}\right)$$

which can be done effectively if the samples u_k are ordered. The methods differ in the sampling of the u_k's. It should be noted that one does not need to deal with the samples $\tilde{x}_i$ or x_k directly; one needs only the weights $\omega^{(i)}$ and determines the indices i.

The resampling schemes listed in Hol et al. [2006] are the following:

1. ***Multinomial sampling***:

   Draw N uniform samples

   $$\tilde{u}_k \sim \mathscr{U}[0,1), \qquad k = 1, \ldots, N,$$

   and generate the N ordered random numbers u_k recursively according to

   $$u_N := \tilde{u}_N^{\frac{1}{N}}, \qquad u_k := u_{k+1} \tilde{u}_k^{\frac{1}{k}} \text{ for } k = N-1, \ldots, 1.$$

2. ***Stratified sampling***:

   Draw N uniform samples

$$\tilde{u}_k \sim \mathscr{U}[0,1), \qquad k = 1,\ldots,N,$$

   and generate the N ordered random numbers u_k according to

$$u_k := \frac{(k-1)+\tilde{u}_k}{N} \quad \text{for all } k = 1,\ldots,N.$$

3. ***Systematic sampling***: Draw one uniform sample $\tilde{u} \in \mathscr{U}[0,1)$ and generate the N ordered random numbers u_k according to

$$u_k := \frac{(k-1)+\tilde{u}}{N} \quad \text{for all } k = 1,\ldots,N.$$

4. ***Residual sampling***: Allocate $n_i' := \lfloor N\omega^{(i)} \rfloor$ copies of the particle $\tilde{x}_i$ to the new distribution. Additionally, resample $m := N - \sum_i n_i'$ particles from $\tilde{x}_i$ where the probability for selecting $\tilde{x}_i$ is proportional to $\omega^{(i)'} := N\omega^{(i)} - n_i'$ using one of the resampling schemes mentioned earlier.

All these algorithms are unbiased and can be implemented in $O(N)$ time. They differ in the complexity of their single steps: Multinomial being the most costly algorithm, followed by stratified sampling and finally systematic sampling. Residual sampling is more difficult to place.

Concerning the variance reduction, it can be shown that stratified and residual sampling have lower variance than multinomial sampling. Due to the fact that systematic sampling produces its samples dependently, it is hard to conduct a proper variance analysis, and in Douc et al. [2005] an artificial example is given where the variance increases. Nevertheless, systematic sampling has to be preferred, which is based on considerations on the ***Koksma-Hlawka inequality*** (Hlawka [1961])

$$|\mathbf{E}^{p_M}[g] - \mathbf{E}^{\hat{p}_N}[g]| \leq D_N^*(u_1,\ldots,u_N) V_{\mathrm{HK}}(g)$$

where $V_{\mathrm{HK}}(g)$ is the total variation in the sense of Hardy and Krause; the star discrepancy D_N^* is defined as

$$D_N^*(\{u_i\}) = \sup_{a\in[0,1)^d} \left| \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{(0,a]}(u_i) - |[0,a)| \right|$$

where $|\cdot|$ denotes volume. Systematic sampling shows the lowest discrepancy and smaller differences between the expected values (cf. Hol et al. [2006]).

Rejection sampling Another method to sample directly from a density $p(x)$ with a proposal function $q(x)$ is ***rejection sampling***, also called ***acceptance-rejection method*** or ***accept-reject algorithm***, which is applicable if an upper bound for $\sup_x p(x)/q(x)$ is known. It is done as follows: To sample from p, we first draw x from a density q, and then accept with probability

$$\frac{p(x)}{Mq(x)},$$

where M is any constant such that

$$M \geq \sup_x \frac{p(x)}{q(x)}.$$

If the generated x is not accepted, this procedure is to be repeated until it is. The densities p and q are needed only up to proportionality.

Markov Chain Monte Carlo (MCMC) methods

We adopt now the presentation in Green [2001]. We first assume again that a suitable reference measure μ is given and express the distributions as densities with respect to this measure. In this way, let p now denote the target density, and P the transition kernel.

The idea of ***Markov Chain Monte Carlo (MCMC) methods*** is:

- Given a density distribution p, produce a Markov chain (X_t) with stationary distribution p.

From the ergodicity theorem, it follows that we can achieve this at least for almost all starting values x if we find a φ-irreducible, aperiodic Markov chain with invariant distribution p.

Detailed balance The key idea in most practical approaches to constructing MCMC methods is ***time-reversibility*** or ***detailed balance***:

$$p(x)P(x,y) = p(y)P(y,x) \qquad \text{for all } x,y \in \mathscr{X}.$$

This is a sufficient but not necessary condition for invariance (see e.g. Green [2001]):

Theorem 3.18: *The distribution p is invariant for P if we have detailed balance (time-reversibility).*

It is far easier to work with detailed balance than with invariance.

In the next paragraphs, we present basic steps for MCMC chains. How these can be build up to obtain a irreducible and aperiodic chain is treated in a subsequent paragraph.

The Gibbs sampler The step of the ***Gibbs sampler*** is very simple: Discard the current value of a single component x_i of x and replace it by a value y_i drawn from the full conditional distribution induced by p:

$$p(x_i \,|\, x_{-i}),$$

(where $-i$ stands for the set of indices $\{j \mid j \neq i\}$), and keeping the current values of the other variables: $y_{-i} = x_{-i}$. We are thus using the kernel:

$$P(x,y) = p(y_i \,|\, x_{-i}) I[x_{-i} = y_{-i}],$$

and detailed balance holds because given x_{-i}, the components x_i and y_i are independent, and identically distributed as $p(x_i \,|\, x_{-i})$. This recipe was named Gibbs sampler by Geman and Geman [1984], whose work brought it to the attention of statisticians, but is known earlier as the "heat bath" by physicists.

The Metropolis method In the ***Metropolis method***, we first find a new candidate or ***proposal*** y by drawing y_i from an arbitrary density (***proposal density***) $q_i(y_i \,|\, x)$ conditioned on x, and setting again $y_{-i} = x_{-i}$. We write this as transition kernel $q_i(x,y) := q_i(y_i \,|\, x)$, and impose the symmetry requirement

$$q_i(x,y) = q_i(y,x).$$

This proposal is then accepted as the next state of the chain with pobability

$$\alpha(x,y) = \min\left(1, \frac{p(y)}{p(x)}\right) = \min\left(1, \frac{p(y_i \,|\, x_{-i})}{p(x_i \,|\, x_{-i})}\right),$$

and otherwise x is left unchanged. The target density p is here needed only up to proportionality, at two values, the current and the proposed next states.

This recipe is due to Metropolis et al. [1953]. We get the Gibbs sampler as a special case if the proposal density $q_i(y_i \,|\, x)$ is just the full conditional $p(y_i \,|\, x_{-i}) = p(x_i \,|\, x_{-i})$, so that the acceptance probability is 1.

The Metropolis-Hastings sampler The ***Metropolis-Hastings sampler*** is an important generalization of the Metropolis method introduced by Hastings [1970], and overlooked by statisticians for nearly 20 years. The symmetry of q is not needed here. The acceptance probability becomes:

$$\alpha(x,y) = \min\left(1, \frac{p(y) q_i(y,x)}{p(x) q_i(x,y)}\right) = \min\left(1, \frac{p(y_i \,|\, x_{-i}) q_i(x_i \,|\, y)}{p(x_i \,|\, x_{-i}) q_i(y_i \,|\, x)}\right).$$

The optimality in some senses of this particular choice of $\alpha(x,y)$ over any other choice preserving detailed balance was demonstrated by Peskun [1973].

The Metropolis method is the special case where q is symmetric.

Proof of detailed balance We still follow Green [2001]. Beause Gibbs sampler and Metropolis method are each special cases of the Metropolis-Hastings method, it suffices to proof detailed balance for this general case. Let $x \neq y$. We need only the fact that

$$\frac{\alpha(x,y)}{\alpha(y,x)} = \frac{p(y)}{p(x)} \frac{q_i(y,x)}{q_i(x,y)}.$$

We thus get immediately if $x \neq y$:

$$p(x)P(x,y) = p(x) q_i(x,y) \alpha(x,y) = p(y) q_i(y,x) \alpha(y,x) = p(y)P(y,x).$$

Full conditionals In each of the samplers it is possible to update several components simultaneously, using the ***full conditionals***

$$p(x_A \,|\, x_{-A})$$

where $x_A := \{x_j \mid j \in A\}$ and $x_{-A} := \{x_j \mid j \notin A\}$. In Gibbs, one has to draw from the full conditionals, in Metropolis and Metropolis-Hastings, one has to evaluate it up to a multiplicative constant at the current and the proposed values. The determination of the full conditionals is very simplified by the use of graphical models.

Combining kernels The detailed balance ensures only that p is invariant, i.e. if $x \sim p$ before the transition, then so it will be afterwards. To ensure that p is also the limiting distribution of the chain (ergodicity), we must combine such kernels to make a Markov chain that is irreducible (and aperiodic). To do this, one has to scan over the several kernels (indexed by i or A) in a way such that each variable is visited often enough. We denote the kernels by $P_1, P_2, \ldots, P_m$. There are two main schemes (Green [2001]):

- ***Cyclic kernel***: Go systematically through the kernels P_i:

$$P := P_1 P_2 \cdots P_m.$$

- ***Mixture kernel***: Go randomly through the kernels P_i with equal probability for each kernel:

$$P := \frac{1}{m} \sum_{i=1}^{m} P_i.$$

Note that the mixture kernel preserves detailed balance, while the cyclic kernel does not. Nevertheless, p remains invariant for both combinations.

Choices for proposal distributions There is a completely free choice of the proposal distribution $q_i(y_i \,|\, x)$. Nevertheless, typically a small number of standard specifications is used (Green [2001]):

- ***Independence Metropolis-Hastings***: Propose the new state y_i independent of x, i.e.

$$q_i(y_i \,|\, x_{-i}) = q_i(y_i).$$

  The acceptance probability then is

$$\alpha(x,y) = \min\left(1, \frac{p(y)q_i(x)}{p(x)q_i(y)}\right).$$

  This choice is of limited use in practice, and more considered for theoretical reasons.

- ***Random walk Metropolis***: If the proposal is given as

$$q_i(x,y) = q_i(\|y_i - x_i\|)$$

then

$$\frac{q_i(y,x)}{q_i(x,y)} = 1,$$

i.e. q_i is symmetric and the acceptance probability simplifies. This proposal amounts to adding a random walk increment $\sim q_i$ to the current x_i.

- ***Random walk on the log-scale***: When a component x_i is necessarily positive, it is convenient to only propose changes which leave y_i positive. Then rather a multiplivative than an additive update is suggested. We then choose to propose an (additive) increment $\sim q_i(x,y)$ to $\log x_i$ instead to x_i, i.e.

$$q_i(x,y) = \tilde{q}_i(|\log y_i - \log x_i|)\left|\frac{\partial x_i}{\partial \log x_i}\right|,$$

and we find

$$\frac{q_i(y,x)}{q_i(x,y)} = \frac{y_i}{x_i}.$$

Comparing Metropolis-Hastings to rejection sampling There is a superficial resemblance between Metropolis-Hastings and rejection sampling (see e.g. Green [2001]):

Recall that rejection sampling is done as follows: To sample from p, we first draw y from a density q, and then accept with probability

$$p(y)/(Mq(y)),$$

where M is any constant such that

$$M \geq \sup_y p(y)/q(y).$$

If the generated y is not accepted, this procedure is to be repeated until it is.

The differences of the Metropolis-Hastings algorithm in comparison with rejction sampling are:

- p/q need not be bounded;
- one has not to repeat if the proposal is rejected;
- one ends up with a Markov chain, not with an independent sequence.

Reversible jump methods

We let follow a generalization of the Metropolis-Hastings scheme due to Green [1995] (we use the presentation given in Green [2001]). The original method of Hastings is already quite general in that it applies to densities $p(x)$ and $q(x,y)$ with respect to an arbitrary reference measure on $\mathscr{X}$. Considering the well-known counting and Lebesgue measures, this covers discrete and continuous distributions in any finite number of dimensions. However, the formulation is a little restrictive for problems where there is no elementary reference measure for the target function. These problems occur most prominently in cases where the dimension of the parameters varies. The more general Metropolis-Hastings method addresses this wider range of problems.

General detailed balance condition If P is a general transition kernel and p its invariant distribution, the ***general detatiled balance condition*** reads as

$$\int_{(x,y)\in A\times B} p(dx)P(x,dy) = \int_{(x,y)\in A\times B} p(dy)P(y,dx)$$

for all measurable sets $A,B\subseteq\mathscr{X}$. According to the Metropolis-Hastings recipe, the kernel P is constructed in two steps: First draw a proposal y from the proposal measure $q(x,dy)$ and then accept it with probability $\alpha(x,y)$. If we reject, we stay in the current state, so that $P(x,dy)$ has an atom at x. This makes an equal contribution to each side of the detailed balance equation and can be neglected. We are thus left with the requirement

$$\int_{(x,y)\in A\times B} p(dx)q(x,dy)\alpha(x,y) = \int_{(x,y)\in A\times B} p(dy)q(y,dx)\alpha(y,x).$$

One in some sense has to "solve" this collection of equations for α. If μ is a *symmetric* measure on $\mathscr{X}\times\mathscr{X}$ such that $p(dx)q(x,dy)$ admits a density f with respect to μ, i.e.

$$f(x,y)\mu(dx,dy) = p(dx)q(x,dy),$$

then the above equation becomes

$$\int_{(x,y)\in A\times B} f(x,y)\alpha(x,y) = \int_{(x,y)\in A\times B} f(y,x)\alpha(y,x),$$

and using the symmetry of μ, this is clearly satisfied for all $A,B\subseteq\mathscr{X}$ such that (A,B) is μ-measurable, if

$$\alpha(x,y)f(x,y) = \alpha(y,x)f(y,x).$$

Thus, similarly to the standard Metropolis-Hastings method, we have the acceptance probability

$$\alpha(x,y) = \min\left(1,\frac{f(y,x)}{f(x,y)}\right).$$

This may formally be written as

$$\alpha(x,y) = \min\left(1,\frac{p(dy)q(y,dx)}{p(dx)q(x,dy)}\right)$$

and shows thus the similarity to the standard acceptence ratio, but this formula makes only sense if the existence of a symmetric reference measure μ is assumed.

In some cases, the reference measure μ may be given explicitly, but in other situations, μ is much less explicit. Then the following construction may be useful which also in the standard Metropolis-Hastings method may provide some simplifications when implementing the recipe.

Explicit representation using random numbers Let $\mathscr{X} \subseteq \mathbb{R}^d$ and let p be a density with respect to the d-dimensional Lebesgue measure λ^d. To sample from the transition density $P(x,y)$, the practical implementation goes via generating a random vector u of dimension r from a known density g, and then forming the sample y by some suitable deterministic function: $y = y(x,u)$. If now the reverse transition is made by a random number $u' \sim g$ and $x = x(y,u')$, if the transformation from (x,u) to (y,u') is a bijection, and if both it and its inverse are differentiable, then by the standard change-of-variable formula, the $(d+r)$-dimensional integral equality in the case of acceptance holds if

$$p(x)g(u)\alpha(x,y) = p(y)g(u')\alpha(y,x)\left|\frac{\partial(y,u')}{\partial(x,u)}\right|,$$

whence a valid choice for α is

$$\alpha(x,y) = \min\left(1, \frac{p(y)g(u')}{p(x)g(u)}\left|\frac{\partial(y,u')}{\partial(x,u)}\right|\right).$$

It is often easier to work with this expression than with the usual one.

MCMC for variable dimension problems There is a huge variety of statistical problems where the number of parameters is not fixed but also subject to inference. Examples are variable selection, mixture estimation, change point analysis, and model selection. We need an MCMC sampler that jumps between parameter spaces of different dimensions. In the general context of this section, this is easily accomplished. Let the state variable x be from a union of spaces of differing dimension d_k:

$$\mathscr{X} = \bigcup_k \mathscr{X}_k.$$

One uses then a range of ***move types*** m, each providing a transition kernel P_m, and requires detailed balance for each:

$$\int_{x\in A} p(dx)P_m(x,B) = \int_{y\in B} p(dy)P_m(y,A)$$

for all measurable sets $A, B \subseteq \mathscr{X}$. The Metropolis-Hastings idea still works here, but it is a bit more difficult to make the acceptance ratio make sense. The proposal measure q is now the joint distribution of move type m and proposed destination y, so for each $x \in \mathscr{X}$,

$$\sum_m \int_{y\in\mathscr{X}} q_m(x,dy) \le 1.$$

(If the inequality is strict, there is positive probability that no move is attempted.) The detailed balance condition becomes

$$\int_{(x,y)\in A\times B} p(dx)q_m(x,dy)\alpha_m(x,y) = \int_{(x,y)\in A\times B} p(dy)q_m(y,dx)\alpha_m(y,x)$$

for all m and measurable A,B. This leads to the formal solution

$$\alpha_m(x,y) = \min\left(1, \frac{p(dy)q_m(y,dx)}{p(dx)q_m(x,dy)}\right)$$

which makes only sense subject to the existence of a symmetric reference measure μ_m for $p(dy)q_m(y,dx)$.

A practical implementation will again use the procedure of the preceding paragraph: We need a differentiable bijection between (x,u) and (y,u'), where u,u' are vectors of random numbers used to go between x and y in each direction. Suppose these have densities $g_m(u\,|\,x)$ and $g_m(u'\,|\,y)$. In the variable dimension context, move type m might use transitions between $\mathscr{X}_{k_1}$ and $\mathscr{X}_{k_2}$; if these spaces have dimensions d_1 and d_2, and p is absolutely continuous with respect to measures ν_{d_1} and ν_{d_2} in the respective spaces, then the dimensions of u and u', r_1 and r_2 say, must satisfy the dimension-balancing condition

$$d_1 + r_1 = d_2 + r_2.$$

We can then write

$$\alpha_m(x,y) = \min\left(1, \frac{p(y)g_m(u')}{p(x)g_m(u)}\left|\frac{\partial(y,u')}{\partial(x,u)}\right|\right).$$

Improving the performance of MCMC methods

Auxiliary variables We follow again Green [2001]. Edwards and Sokal [1988] proposed a way to improve mixing by augmenting the state space so that the original target appears as the marginal equilibrium distribution. Starting from $p(x)$, introduce some additional variables u, with $p(u\,|\,x)$ arbitrarily chosen. Then the joint distribution is

$$p(x,u) = p(x)p(u\,|\,x),$$

for which $p(x)$ is certainly the marginal for x.

The slice sampler One application of auxiliary variables is the ***slice sampler***. Suppose $p(x)$ factorizes as

$$p(x) = p_0(x)b(x)$$

where $p_0(x)$ is a (possibly unnormalized) distribution that is easy to sample from, and $b(x)$ is the awkward part, often representing the "interactions" between variables that are slowing down the chain. Take a one-dimensional u with $u\,|\,x \sim \mathscr{U}[0,b(x)]$. Then

$$p(x,u) = p(x)p(u\,|\,x) = p_0(x)b(x)\frac{\mathbf{1}_{[0,b(x)]}(u)}{b(x)}$$

so that

$$p(x|u) \propto p_0(x)$$

restricted to (conditional on) the event $\{x \mid b(x) \geq u\}$. At least when this $p(x|u)$ can be sampled without rejection, we can easily implement a Gibbs sampler, drawing u and x in turn.

3.6 State space systems and recursive computations

The MCMC methods of the last section are general purpose tools for Bayesian analysis. They work well for static models. They are not suited in the dynamical setting, especially if the systems evolve over a long time period. Our aim in this section is to explain the Sequential Monte Carlo (SMC) methods based on recursive formulations of the filtering densities of state space systems.

History of SMC methods We follow Doucet et al. [2000]. From the mid 1960's, a great deal of attention was devoted to the approximation of filtering distributions, see e.g. Jazwinski [1970]. The most popular examples are the Extended Kalman Filter and the Gaussian sum filter (based on analytical approximations, Anderson and Moore [1979]). During 1960's and 1970's, sequential MC integration methods were used in the automatic control field (Akashi and Kumamoto [1975], Handschin and Mayne [1969], Handschin [1970], Zaritskii et al. [1975]). Possibly due to computational limitations, these methods were largely neglected. Only in the late 1980's, the massive increase in computational powers allowed the rebirth of numerical integration methods for Bayesian filtering (Kitagawa [1987]). Research from the 1990's on changed the focus to Monte Carlo integration methods (Müller [1992], West [1993], Gordon et al. [1993], Kong et al. [1994], Liu and Chen [1998]).

3.6.1 General state space models

Let $(\mathscr{X}, \mathfrak{A}_{\mathscr{X}}, \mu)$ and $(\mathscr{Y}, \mathfrak{A}_{\mathscr{Y}}, \nu)$ be two probability spaces, i.e.

- $\mathscr{X}$ and $\mathscr{Y}$ are sets,
- $\mathfrak{A}_{\mathscr{X}}$ and $\mathfrak{A}_{\mathscr{Y}}$ are σ-algebras on $\mathscr{X}$ and $\mathscr{Y}$, respectively,
- μ and ν are (reference) probability measures on $\mathfrak{A}_{\mathscr{X}}$ and $\mathfrak{A}_{\mathscr{Y}}$, respectively.

For each $t \in \mathbb{N}$ let X_t be a random variable on $\mathscr{X}$, i.e. X_t is a function

$$X_t : \mathscr{X} \longrightarrow \mathbb{R}$$

such that for all $r \in \mathbb{R}$ the set $A_r := \{x \in \mathscr{X} \mid X_t(x) \leq r\}$ is $\mathfrak{A}_{\mathscr{X}}$-measurable, i.e. $A_r \in \mathfrak{A}_{\mathscr{X}}$. Similarly, for each t let $Y_t : \mathscr{Y} \longrightarrow \mathbb{R}$ be a random variable. We use the following notation: For every $s \leq t$ we write $X_{s:t}$ for $(X_s, X_{s+1}, \ldots, X_t)$ and similarly $Y_{s:t}$ for $(Y_s, Y_{s+1}, \ldots, Y_t)$.

We follow the presentation of Künsch [2001] without adapting his notation. The ***general state space model*** consists of

- the unobserved state sequence $(X_t)_{t\in\mathbb{N}}$,
- the observation sequence $(Y_t)_{t\in\mathbb{N}\setminus\{0\}}$,

with the following properties:

- ***State evolution***: $X_0, X_1, X_2, \ldots$ is a Markov chain with
  - Initial distribution $\Pr(X_0 \in dx) =: f_0(x)d\mu(x)$, and
  - Transition distribution $\Pr(X_t \in dx \,|\, X_{t-1} = x_{t-1}) =: f_t(x \,|\, x_{t-1})d\mu(x)$.
- ***Generation of observations***:
  - Conditionally on X_t, each Y_t is independent of Y_s and X_s for all $s \neq t$, and
  - Y_t depends on X_t through the observation distribution

$$\Pr(Y_t \in dy \,|\, X_t = x_t) =: g_t(y \,|\, x_t)d\nu(y).$$

Thus, to determine an individual state space model completely, one has to provide the following densities:

- Initial state density (X_0): $f_0(x)$
- State transition density ($X_{t-1} \to X_t$): $f_t(x \,|\, x_{t-1})$
- Observation transition density ($X_t \to Y_t$): $g_t(y \,|\, x_t)$

It should be remarked that we explicitly assumed absolutely continuous measures with respect to μ and ν for the transition distributions of states and observations, respectively (i.e. there exist densities in both cases). We will later come back to the case where the measures are not absolutely continuous.

The general state space model pictured as a graphical model looks like:

$$\begin{array}{ccccccccccc} X_0 & \longrightarrow & X_1 & \longrightarrow & X_2 & \longrightarrow & \ldots & \longrightarrow & X_t & \longrightarrow & X_{t+1} & \longrightarrow & \ldots \\ & & \downarrow & & \downarrow & & & & \downarrow & & \downarrow & & \\ & & Y_1 & & Y_2 & & \ldots & & Y_t & & Y_{t+1} & & \ldots \end{array}$$

Usually, $\mathscr{X}$ and $\mathscr{Y}$ are finite vector spaces, e.g. $\mathbb{R}^n$. If $\mathscr{X}$ is discrete, then the general state space model is called ***Hidden Markov Model (HMM)***, but sometimes the two terms are used as synonyms.

Alternatively, the state space model can be expressed by

$$X_t = F_t(X_{t-1}, V_{t-1}), \qquad Y_t = G_t(X_t, W_t)$$

where F_t and G_t are arbitrary deterministic functions, and where $(V_t)_{t\in\mathbb{N}}$ and $(W_t)_{t\in\mathbb{N}^*}$ are two independent stochastic processes consisting of ***white noise***, i.e. each random variable V_{t_1} is independent of each other V_{t_2} for $t_1 \neq t_2$, and analoguously for W_t.

Examples of state space systems

Linear state space model A special case is the ***linear model***

$$X_t = F_t X_{t-1} + V_{t-1}, \qquad Y_t = G_t X_t + W_t$$

where $\mathscr{X}$ and $\mathscr{Y}$ are vector spaces and F_t and G_t are linear. If additionally V_t and W_t are Gaussian white noise, then the model is a ***normal (or Gaussian) linear model***.

As shown in subsection 3.4.2, a stationary Gaussian ARMA(p,q) process $(Y_t)_{t\in\mathbb{N}}$ can be represented as a linear general state space model by defining a $r = \max(p, q+1)$-dimensional state vector x_t. This can be extended to generalizations of the ARMA-models like ARIMA (Autoregressive Integrated Moving Average) models, which incorporate an additional integral term.

Modelling of outliers In classical identification procedures, most algorithms need the assumption of normal distributions of the data. This assumption is often not fulfilled with real data which results e.g. in observed values which are far more away from the expected value than the variance of the Gaussian distribution would suggest. These values are called ***outliers***. To force the data to obey the Gaussian distribution, it is usual to preprocess the measured data, one step of this preprocessing being the removal of outliers. This often is done by hand. In the classical case, this is a necessary procedure: Outliers divagate from the usually assumed normal distribution of the errors and lead to biased results. A better idea may be to include the possibility of outliers into the model. This is done by choosing a distribution which is more heavy-tailed than the normal distribution, for example a mixture of normal distributions, a Student t-distribution, or so-called α-stable distributions (for modelling and simulation with α-stable noise see Lombardi and Godsill [2004]).

Stochastic volatility models Another example is the ***stochastic volatility model***, in its simplest form given by (see Künsch [2001])

$$\begin{aligned} X_t &= m + \phi X_{t-1} + V_{t-1}, \\ Y_t &= \exp(X_t) W_t, \end{aligned}$$

with two independent Gaussian white noises (V_t) and (W_t). The task is to estimate the parameters (m, ϕ, σ_V^2) and to make predictions about the occurrence of large negative values of Y_t for assessing risk (for further details see Shephard [1996]).

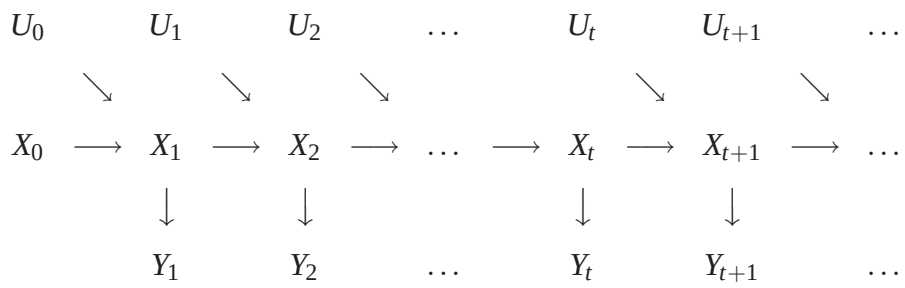
Stochastic differential equations with discrete observations Examples of general state space models can also be obtained from stochastic differential equations with discrete observations (see e.g. Cérou et al. [2000]):

Let $\{X_t, t \in \mathbb{R}_{\geq 0}\}$ be a continuous-time ***diffusion process***, i.e. given by the solution of the stochastic differential equation

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t, \qquad X_0 \sim \mu_0(dx),$$

with Brownian motion $\{W_t, t \in \mathbb{R}_{\geq 0}\}$. Then a Markov chain X_{t_k} is obtained by sampling this process at discrete time points t_k, $k \in \mathbb{N}$. Transition from time t_k to time t_{k+1} of this Markov chain is thus given implicitly by the stochastic differential equation. The state sequence is not observed, but instead an observation sequence $\{Y_{t_k}, k \in \mathbb{N}\}$ is available.

State space models with control Several extensions of state space models are possible, e.g. state space models with control, where U_t denotes the controls (inputs):



Many other examples can be found in

- finance applications: stochastic volatility models (Pitt and Shephard [1999]),
- biology: ion channels (De Gunst et al. [2001]), DNA and protein sequences (Thompson [2001], Durbin et al. [1998]),
- engineering: target tracking (Gordon et al. [1993]), control (Whittle [1996]), speech analysis (Rabiner and Juang [1993]), digital enhancement of speech and audio signals (Godsill and Rayner [1998]), blind deconvolution of digital communication channels (Clapp and Godsill [1999], Liu and Chen [1995]), position estimation of mobile robots (Fox et al. [2001]),
- geophysics: weather prediction (Hughes et al. [1999]).

3.6.2 Filtering and smoothing

The main tasks to do are (Künsch [2001]):

- Inference about the states on observed data $y_{s:t}$ for a given model (f_t and g_t known).
- Inference about the unknown parameters in f_t and g_t.

Inference about a state X_s given $y_{1:t}$ is called

- ***prediction*** if $s > t$,
- ***filtering*** if $s = t$,
- ***smoothing*** if $s < t$.

There is an increasing difficulty:

Prediction $\longrightarrow$ Filtering $\longrightarrow$ Smoothing $\longrightarrow$ Parameter Estimation

In the following, we will be confronted with an abundance of different probability densities. We therefore adopt the usual sloppy notations involving the same notation for different densities. Which density is actually meant will be clear from the arguments: For example, we write $p(x_{0:t}, y_{1:t})$ for the joint density of states and observations

$$p_{0:t,1:t}(x_{0:t}, y_{1:t}),$$

and $p(x_{0:t} \,|\, y_{1:t})$ for the joint smoothing density

$$p_{0:t\,|\,1:t}(x_{0:t} \,|\, y_{1:t}),$$

as well as $p(x_s \,|\, y_{1:t})$ for the marginal filtering and smoothing densities

$$p_{s\,|\,1:t}(x_s \,|\, y_{1:t}).$$

We use the notation p also for densities involving only observations, e.g. $p(y_t \,|\, y_{1:t-1})$ denoting the density for the observation y_t given $y_{1:t-1}$.

Non-recursive formulation of densities

For the following formulas we use Künsch [2001].

Joint smoothing density The density where the main interest is in is the ***joint smoothing density***, i.e. the conditional density of the states given all observations $y_{1:t}$. It is given by:

$$p(x_{0:t} \,|\, y_{1:t}) = \frac{p(x_{0:t}, y_{1:t})}{p(y_{1:t})}$$

where the joint density of states and observations $(X_{0:t}, Y_{1:t})$ is given by

$$p(x_{0:t}, y_{1:t}) = f_0(x_0) \prod_{s=1}^{t} f_s(x_s \,|\, x_{s-1}) g_s(y_s \,|\, x_s),$$

and where the joint density of the observations

$$p(y_{1:t}) = \int \cdots \int p(x_{0:t}, y_{1:t}) d\mu(x_0) \cdots d\mu(x_t)$$

which is a constant (the y_s's are fixed!) is obtained by marginalization of $p(x_{0:t}, y_{1:t})$. An analytical computation of this high-dimensional integral is practically impossible.

The joint process $(X_{0:t}, Y_{1:t})$ is again a Markov process, and the joint smoothing density being proportional to the joint density

$$p(x_{0:t} \,|\, y_{1:t}) \propto p(x_{0:t}, y_{1:t})$$

implies that conditionally on $y_{1:t}$ the state variables are also Markovian. This will open the door to recursive formulations of the joint smoothing density.

Marginal filtering and smoothing densities Apart from the joint smoothing density, there are two smoothing distributions of interest (see e.g. Doucet et al. [2000]):

- ***Fixed-lag smoothing***: For a fixed lag $L \in \mathcal{N}$, the fixed-lag distribution is given by

$$p(x_{t-L} \,|\, y_{1:t}).$$

- ***Fixed-interval smoothing***: Here, one is interested in all ***(marginal) smoothing densities***

$$p_{s\,|\,1:t}(x_s \,|\, y_{1:t}) \qquad \text{for all } s = 0, \ldots, t-1$$

  and the ***filtering density***

$$p_{t\,|\,1:t}(x_t \,|\, y_{1:t}).$$

The (marginal) filtering and smoothing densities $p_{s\,|\,1:t}$ can be obtained by marginalization from the joint smoothing density:

$$p_{s\,|\,1:t}(x_s \,|\, y_{1:t}) = \int \cdots \int p(x_{0:t} \,|\, y_{1:t}) d\mu(x_0) \cdots \widehat{d\mu(x_s)} \cdots d\mu(x_t)$$

if $s \leq t$, where $\widehat{d\mu(x_s)}$ means: leave this integration out. Here again, these integrals are practically not computable.

Recursive formulas

The way out of the computational difficulties is the use of a recursive procedure to break down the high-dimensional integrals into a series of lower-dimensional ones. We follow again Künsch [2001].

Prediction of states The prediction densities for the states can be computed from the filtering densities according to the following recursion in k:

$$p_{t+k\,|\,1:t}(x_{t+k} \,|\, y_{1:t}) = \int p_{t+k-1\,|\,1:t}(x \,|\, y_{1:t})\, f_{t+k}(x_{t+k} \,|\, x) d\mu(x).$$

Prediction of observations The prediction densities for the observations can be computed from the prediction densities for the states according to the following recursion in k:

$$p(y_{t+k} \,|\, y_{1:t}) = \int p_{t+k\,|\,1:t}(x \,|\, y_{1:t}) g_{t+k}(y_{t+k} \,|\, x) d\mu(x).$$

Joint observation density As a byproduct one obtains also the joint density of the observations, because

$$p(y_{1:t}) = \prod_{s=1}^{t} p(y_s \,|\, y_{1:s-1})$$

and $p(y_s \,|\, y_{1:s-1})$ is given by the above recursion formula for the prediction densities of the observations.

Joint smoothing density The recursions for the state sequence given the observations (joint smoothing density) $p(x_{0:t}\,|\,y_{1:t}) \propto p(x_{0:t}, y_{1:t})$ can be obtained in two steps:

(1) Propagation:

$$p(x_{0:t}\,|\,y_{1:t-1}) = p(x_{0:t-1}\,|\,y_{1:t-1})\, f_t(x_t\,|\,x_{t-1}).$$

The proof uses total probability and that x_t is conditionally independent from $y_{1:t-1}$ given x_{t-1}.

(2) Update:

$$p(x_{0:t}\,|\,y_{1:t}) = \frac{p(x_{0:t}\,|\,y_{1:t-1}) g_t(y_t\,|\,x_t)}{p(y_t\,|\,y_{1:t-1})}.$$

The proof uses Bayes Rule and that y_t is conditionally independent from $y_{1:t-1}$ given x_t.

This combined gives:

$$\begin{aligned} p(x_{0:t}\,|\,y_{1:t}) &= \frac{p(x_{0:t-1}\,|\,y_{1:t-1})\, f_t(x_t\,|\,x_{t-1}) g_t(y_t\,|\,x_t)}{p(y_t\,|\,y_{1:t-1})} \\ &\propto p(x_{0:t-1}\,|\,y_{1:t-1})\, f_t(x_t\,|\,x_{t-1}) g_t(y_t\,|\,x_t) \end{aligned}$$

where the normalization constant $p(y_t\,|\,y_{1:t-1})$ may as well be computed recursively as prediction density of the observations (see above):

$$p(y_t\,|\,y_{1:t-1}) = \int p_{t\,|\,1:t-1}(x\,|\,y_{1:t-1}) g_t(y_t\,|\,x) d\mu(x).$$

Recursive filtering The recursion for the filtering densities is obtained by marginalization:

Propagation: From filter density to prediction density

$$p_{t\,|\,1:t-1}(x_t\,|\,y_{1:t-1}) = \int p_{t-1\,|\,1:t-1}(x\,|\,y_{1:t-1})\, f_t(x_t\,|\,x) d\mu(x).$$

Similarly: $p_{s+1\,|\,1:t}$ from $p_{s\,|\,1:t}$ for all $s > t$.

Update: From prediction density to filter density

$$p_{t\,|\,1:t}(x_t\,|\,y_{1:t}) = \frac{p_{t\,|\,1:t-1}(x_t\,|\,y_{1:t-1}) g_t(y_t\,|\,x_t)}{p(y_t\,|\,y_{1:t-1})} \propto p_{t\,|\,1:t-1}(x_t\,|\,y_{1:t-1}) g_t(y_t\,|\,x_t).$$

Propagation and update combined leads to:

$$\begin{aligned} p_{t\,|\,1:t}(x_t\,|\,y_{1:t}) &= \frac{\int p_{t-1\,|\,1:t-1}(x\,|\,y_{1:t-1})\, f_t(x_t\,|\,x) d\mu(x) g_t(y_t\,|\,x_t)}{p(y_t\,|\,y_{1:t-1})} \\ &\propto \int p_{t-1\,|\,1:t-1}(x\,|\,y_{1:t-1})\, f_t(x_t\,|\,x) d\mu(x) g_t(y_t\,|\,x_t). \end{aligned}$$

Recursions for the marginal smoothing densities For computing the marginal smoothing densities $p_{s|1:t}$ we have two possibilities; both use the filtering densities $p_{s|1:s}$, but the first method uses additionally the one-step prediction densities for the states, whereas the second method uses the one-step prediction densities for the observations, which is easier to store, because the observations are fixed and these densities therefore just numbers.

Marginal smoothing densities, first method For the first method, we use that given $y_{1:t}$ the state sequence is still a Markov chain. We have the following forward and backward transitions for this chain (see Künsch [2001]):

Forward Transition:

$$\begin{aligned} p(x_s|x_{s-1},y_{1:t}) &= p(x_s|x_{s-1},y_{s:t}) \\ &= \frac{f_s(x_s|x_{s-1})g_s(y_s|x_s)p(y_{s+1:t}|x_s)}{p(y_{s:t}|x_{s-1})}. \end{aligned}$$

where

$$p(y_{s:t}|x_{s-1}) = \int f_s(x_s|x_{s-1})g_s(y_s|x_s)p(y_{s+1:t}|x_s)d\mu(x_s).$$

Backward Transition:

$$\begin{aligned} p(x_s|x_{s+1},y_{1:t}) &= p(x_s|x_{s+1},y_{1:s}) \\ &= \frac{f_{s+1}(x_{s+1}|x_s)p_{s|1:s}(x_s|y_{1:s})}{p_{s+1|1:s}(x_{s+1}|y_{1:s})}. \end{aligned}$$

With the backward transitions, the smoothing densities $p_{s|1:t}$ can be computed by the backward recursion (starting with $p_{t|1:t}$):

$$p_{s|1:t}(x_s|y_{1:t}) = p_{s|1:s}(x_s|y_{1:s}) \int \frac{f_{s+1}(x|x_s)}{p_{s+1|1:s}(x|y_{1:s})} p_{s+1|1:t}(x|y_{1:t})d\mu(x).$$

Marginal smoothing densities, second method The smoothing marginal $p_{s|1:t}$ can also be obtained from $p_{s|1:s}$ by incorporating the additional information $y_{s+1:t}$ using Bayes formula and the fact that $y_{1:s}$ is conditionally independent from $y_{s+1:t}$ given x_s (see Künsch [2001]):

$$\begin{aligned} p_{s|1:t}(x_s|y_{1:t}) &= \frac{p(y_{s+1:t}|x_s,y_{1:s})p(x_s|y_{1:s})}{p(y_{s+1:t}|y_{1:s})} \\ &= \frac{p(y_{s+1:t}|x_s)}{p(y_{s+1:t}|y_{1:s})} p_{s|1:s}(x_s|y_{1:s}). \end{aligned}$$

The ratio occurring in the last row

$$r_{s|t}(x_s|y_{1:t}) := \frac{p(y_{s+1:t}|x_s)}{p(y_{s+1:t}|y_{1:s})} = \frac{p_{s|1:t}(x_s|y_{1:t})}{p_{s|1:s}(x_s|y_{1:s})}$$

is given by the backward recursion (beginning with $r_{t|t} \equiv 1$)

$$r_{s-1|t}(x_{s-1}|y_{1:t}) = \frac{\int f_s(x_s|x_{s-1}) g_s(y_s|x_s) r_{s|t}(x_s|y_{1:t}) d\mu(x_s)}{p(y_s|y_{1:s-1})}.$$

This ratio $r_{s|t}$ is also useful for the forward transitions of the conditional state sequence:

$$p(x_s|x_{s-1}, y_{1:t}) = f_s(x_s|x_{s-1}) g_s(y_s|x_s) \frac{p(y_{s+1:t}|x_s)}{p(y_{s:t}|x_{s-1})},$$

because the fraction in the last row can be expressed as:

$$\frac{p(y_{s+1:t}|x_s)}{p(y_{s:t}|x_{s-1})} = \frac{r_{s|t}(x_s|y_{1:t})}{r_{s-1|t}(x_{s-1}|y_{1:t})} \frac{1}{p(y_s|y_{s-1})}.$$

Thus we get as computation for the low-dimensional marginals of the smoothing distribution:

- Compute $p_{s|1:s}$ and $p(y_s|y_{1:s-1})$ by a forward filter recursion.
- Compute $r_{s|t}$ by a backward recursion.

Derivation of the recursion formulas The above recursion formulas can be directly derived via Bayes' rule and using the various Markov properties. Another approach goes via the reference probability method, where one first considers the case where the states and observations are independent so that conditional expectations are easy to compute, and then to obtain the dependent case by an absolutely continuous change of measure (see Künsch [2001]). The advantage of this approach is that it generalizes to the time continuous case.

Transitions that are not absolutely continuous In the above setting, we assumed absolutely continuous measures for both the state transitions, $f_t(x|x_{t-1})d\mu(x)$, and the transitions from the state to the observations, $g_t(y|x_t)d\nu(y)$. It is easy to see that for filtering, we do not need to assume the state transitions to have densities. If (X_t) is a Markov chain with transition kernels $f_t(dx_t|x_{t-1})$ and if we denote the conditional distributions of X_t given $y_{1:t}$ by $\nu_t(dx_t|y_{1:t})$, then the following recursion holds:

$$\nu_t(dx_t|y_{1:t}) \propto g_t(y_t|x_t) \int \nu_{t-1}(dx|y_{1:t-1}) f_t(dx_t|x).$$

But when densities for the conditional distributions of Y_t given x_t do not exist, then there is no general filtering formula. The case is even worse for smoothing: If the state transitions are not absolutely continuous, then there is no general formula for smoothing. However, in most practical cases where no densities exist, it is nevertheless easy to modify the filtering and smoothing recursions in a straightforward way, e.g. if conditional to x_t, observations Y_t and states X_t are restricted to a simple lower dimensional subspace (often a linear subspace) in which case the conditional distributions restricted to this subspace are absolutely continuous. Another example occurs when a second order Markov chain is converted to a first order model with state $Z_t := (X_{t-1}, X_t)$; distributions are then never absolutely continuous. A simple way out is then to proceed in steps of size two: the transition distributions are then absolutely continuous if the conditional distribution of X_t given (x_{t-1}, x_{t-2}) is (see also Künsch [2001]).

Parameter estimation

We assume now that f_t and g_t depend on a finite-dimensional parameter θ. Then the ***likelihood*** of θ given the observed sequence $y_{1:t}$ is (see Künsch [2001])

$$p(y_{1:t} \,|\, \theta) = \prod_{s=1}^{t} p(y_s \,|\, y_{1:s-1}, \theta).$$

Each factor is obtained as a normalization during the filter recursions (recursive prediction).

The frequentist principle for parameter estimation is to use the ***maximum likelihood (ML) estimator***: Take that θ which maximizes this likelihood. We may use a general purpose optimization algorithm.

Expectation Maximization (EM) algorithm The ***expectation maximization (EM) algorithm*** was developed by Dempster et al. [1977] as a generalization of several special instances, for example one instance is the Baum-Welch algorithm (introduced in Baum et al. [1970]) for finite state Hidden Markov Models (where the forward–backward algorithm of the next subsection is a part of). It consists of an iteration over an E- and an M-step. Let $\theta^{(i)}$ be the approximation of the ML estimator in the i-th step, then

E-step: Compute

$$Q(\theta, \theta^{(i-1)}) = \mathbf{E}\big[\log p(x_{0:t}, y_{1:t}; \theta) \,|\, y_{1:t}, \theta^{(i-1)}\big].$$

M-step: Maximize $Q(\theta, \theta^{(i-1)})$ with respect to θ:

$$\theta^{(i)} := \arg\max Q(\theta, \theta^{(i-1)}).$$

It can be shown that in each iteration, the likelihood increases, but it cannot be guaranteed that the algorithm converges to the global maximum.

Bayesian estimation In the Bayesian viewpoint, there is no conceptual difference between states and parameters; thus the parameters can be incorporated into the states and jointly estimated via the filtering and smoothing densities. A-posteriori densities of the parameters can afterwards be obtained by marginalization. Every further inference is then based on these a-posteriori densities.

3.6.3 Exact algorithms for filtering and smoothing

Analytical computations of the filtering and smoothing densities are only possible in very few cases, practically only in the two following ones:

- the finite-state space (discrete) case, and
- the linear normal case.

We follow again Künsch [2001].

Discrete models

Let X_t be discrete with M possible outcomes, $\#\mathscr{X} = M$, say $\mathscr{X} = \{1,2,\ldots,M\}$. Then the integrals are sums, and we can use the forward-backward algorithm of Baum & Welch:

- Filter recursions (row vector $\times$ matrix, componentwise multiplication of two vectors):

$$p_{t\,|\,1:t}(j\,|\,y_{1:t}) \propto \Big[\sum_{k=1}^{M} p_{t-1\,|\,1:t-1}(k\,|\,y_{1:t-1})\, f_t(j\,|\,k)\Big] g_t(y_t\,|\,j).$$

- Recursions for $r_{s\,|\,t}$:

$$r_{s-1\,|\,t}(j,y_{1:t}) = \frac{1}{p(y_s\,|\,y_{1:s-1})} \sum_{k=1}^{M} f_s(k\,|\,j) g_s(y_s\,|\,k) r_{s\,|\,t}(k,y_{1:t}).$$

The complexity for each step is $O(M^2)$, thus the complexity of the whole algorithm $O(TM^2)$ (if T is the number of observations).

Furthermore, the most likely state sequence can be computed with the Viterbi algorithm (Viterbi [1967]).

Linear normal models

Consider the linear normal model

$$X_t = F_t X_{t-1} + V_{t-1}, \qquad Y_t = G_t X_t + W_t$$

where V_t, W_t is Gaussian white noise. Then all $p_{s\,|\,1:t}$ are Gaussian with mean $m_{s\,|\,t}$ and covariance matrices $R_{s\,|\,t}$. These can be computed by the general recursion with the additional use of some linear algebra. This results in the ***Kalman filter*** (Künsch [2001]):

$$\begin{aligned}
m_{t\,|\,t-1} &= F_t m_{t-1\,|\,t-1},\\
m_{t\,|\,t} &= m_{t\,|\,t-1} + K_t(y_t - G_t m_{t\,|\,t-1}),\\
R_{t\,|\,t-1} &= \mathbf{E}[V_{t-1}V_{t-1}^\top] + F_t R_{t-1\,|\,t-1} F_t^\top,\\
R_{t\,|\,t} &= R_{t\,|\,t-1} - K_t G_t R_{t\,|\,t-1}
\end{aligned}$$

with the ***Kalman gain matrix***

$$K_t = R_{t\,|\,t-1} G_t^\top \left(\mathbf{E}[W_t W_t^\top] + G_t R_{t\,|\,t-1} G_t^\top\right)^{-1}.$$

Here, $\mathbf{E}[V_t V_t^\top]$ and $\mathbf{E}[W_t W_t^\top]$ equal the covariance matrices of the state and the observation noise, respectively.

Similarly, we get the ***Kalman smoother***. The smoothing means and variances are:

$$\begin{aligned}
m_{s\,|\,t} &= m_{s\,|\,s} + \overline{K}_{s+1}(m_{s+1\,|\,t} - m_{s+1\,|\,s}),\\
R_{s\,|\,t} &= R_{s\,|\,s} - \overline{K}_{s+1}(R_{s+1\,|\,s} - R_{s+1\,|\,t})\overline{K}_{s+1}^\top
\end{aligned}$$

with

$$\overline{K}_{s+1} = R_{s\,|\,s} F_{s+1}^\top R_{s+1\,|\,s}^{-1}.$$

Many equivalent versions exist with numerical differences in speed and accuracy.

3.6.4 Approximations

In practically all other cases, computations are difficult. In linear models with non-Gaussian noise, the Kalman mean is still the best *linear* unbiased estimator for the states, but nonlinear estimators can be much better.

Possibilities for approximations of the model densities are:

- Approximation by a Gaussian distribution through linearization (Extended Kalman Filter).
- Approximation by a mixture of Gaussian distributions.
- Approximation by empirical densities (Sequential Monte Carlo).

Extended Kalman filter

In engineering, the most popular approximation for the filter densities of nonlinear/non-Gaussian systems is the ***Extended Kalman Filter (EKF)***. It is constructed by linearization of the (nonlinear) system and following application of the Kalman filter. For instance, the linearization of the state transition is (see Künsch [2001])

$$\begin{aligned} X_t &= F_t(X_{t-1}, V_{t-1}) \\ &\approx F_t(m_{t-1|t-1}, 0) \\ &\quad + \frac{\partial F_t(x,u)}{\partial x}\Big|_{(m_{t-1|t-1},0)} (X_{t-1} - m_{t-1|t-1}) + \frac{\partial F_t(x,u)}{\partial u}\Big|_{(m_{t-1|t-1},0)} V_{t-1}, \\ Y_t &= G_t(X_t, W_t) \\ &\approx G_t(m_{t|t-1}, 0) + \frac{\partial G_t(x,v)}{\partial x}\Big|_{(m_{t|t-1},0)} (X_t - m_{t|t-1}) + \frac{\partial G_t(x,v)}{\partial v}\Big|_{(m_{t|t-1},0)} W_t. \end{aligned}$$

In some cases the extended Kalman filter works well, but (see e.g. Künsch [2001]):

- there are some important cases where it does not work,
- error bounds are extremely difficult to produce,
- the error cannot be reduced by a better (and more complicated) approximation,
- the method yields no information on the conditional distributions which can be very non-Gaussian.

An alternative method is numerical integration, but, as we have already mentioned in the previous section, this is problematic in high dimensions, because the convergence is slow and because it is difficult to construct a reasonable grid in advance.

Another alternative are Monte Carlo methods. We consider first the MCMC methods introduced in the previous section.

MCMC methods

The (joint) smoothing distribution $p(x_{0:t}\,|\,y_{1:t})$ is known up to normalization. Thus, standard Markov Chain Monte Carlo methods can be used to simulate (sample) from this distribution. Important to note is that these methods are only applicable for off-line problems. Recursive (on-line) implementation is not possible. But, in most applications, the MCMC chain mixes extremely slowly, because: If we know x_{t-1}, y_t and x_{t+1} then X_t is determined almost completely. Thus, the changes at each step are too small (cf. Künsch [2001]).

Bayesian inference on parameters With MCMC methods, Bayesian inference about the states and unknown parameters at the same time is easy: Sample iteratively between

$$p(\theta\,|\,x_{0:t},y_{1:t}) \quad \text{and} \quad p(x_{0:t}\,|\,\theta,y_{1:t}).$$

But then, the convergence can be even slower.

Sequential Monte Carlo (SMC): Particle filters

Alternative methods are the ***Sequential Monte Carlo (SMC)*** methods. With these methods, one does not try to approximate the complete high-dimensional target distribution at once; the general idea is rather to decompose the target distribution $p(x)$ into a sequence of distributions $p_t(x)$, $t = 0,\ldots,T$:

$$p_0(x), p_1(x), p_2(x), \ldots, p_T(x) = p(x),$$

such that it is easy to compute the starting distribution $p_0(x)$ and to compute the transitions from $p_t(x)$ to $p_{t+1}(x)$. In the setting of state space systems, these decompositions are naturally given by the joint smoothing densities over the time t,

$$p_t(x) := p(x_{0:t}\,|\,y_{1:t})$$

and the transitions are provided by the recursion formulas described previously. These methods are naturally applicable also for on-line problems.

Whereas the SMC methods originally have been developed for the state space setting where in each (time) step t the dimension of the intermediate distributions increases, the SMC methods have been recently shown to be applicable also in the "static" case, where the intermediate distributions $p_t(x)$ are all defined in the same space and are chosen such that they form an approximation sequence to the target $p(x)$ with increasing complexity. SMC methods in the static context are called ***Sequential Monte Carlo Samplers***, and are starting to be a promising alternative to MCMC methods, with the advantage that the obtained samples can be used from the beginning: there is no burn-in period as with MCMC. This avoids all the difficulties encountered with this, like the need to determine when the Markov chain has converged and similar problems. For further details see Del Moral et al. [2006] or Fearnhead [in press].

We follow Cappé et al. [2007] and Doucet et al. [2000].

Sequential Importance Sampling (SIS) To be able to sample from the joint smoothing density $p(x_{0:t}\,|\,y_{1:t})$, one could try to use importance sampling with a suitably chosen importance distribution $q_{0:t}(x_{0:t}\,|\,y_{1:t})$. Conceptually, this is done by sampling N ***particle paths*** $\tilde{x}_{0:t}^{(i)}$, $i = 1,\ldots,N$, and computing the unnormalized importance weights

$$\tilde{\omega}_t^{(i)} := \frac{p(\tilde{x}_{0:t}^{(i)}\,|\,y_{1:t})}{q_{0:t}(\tilde{x}_{0:t}^{(i)}\,|\,y_{1:t})}$$

for $i = 1,\ldots,N$. The weighted samples $(\tilde{x}_{0:t}^{(i)}, \tilde{\omega}_t^{(i)})$, $1,\ldots,N$, constitute then an approximation to the density $p(x_{0:t}\,|\,y_{1:t})$ via

$$p(x_{0:t}\,|\,y_{1:t})\mu(dx) \approx \sum_{i=1}^{N} \tilde{\omega}_t^{(i)} \delta_{\tilde{x}_{0:t}^{(i)}}(dx).$$

To be able to do this recursively, one has also to define the importance distribution $q_{0:t}(x_{0:t}\,|\,y_{1:t})$ in a recursive way:

$$q_{0:t}(x_{0:t}\,|\,y_{1:t}) := q_{0:t-1}(x_{0:t-1}\,|\,y_{1:t-1})\,q_t(x_t\,|\,x_{t-1}, y_t).$$

From the recursion formulas for the joint smoothing densities (section 3.6.2) it follows that the unnormalized importance weights then take the form

$$\tilde{\omega}_t^{(i)} \propto \tilde{\omega}_{t-1}^{(i)} \times \frac{f_t(\tilde{x}_t^{(i)}\,|\,\tilde{x}_{t-1}^{(i)})\,g_t(y_t\,|\,\tilde{x}_t^{(i)})}{q_t(\tilde{x}_t^{(i)}\,|\,\tilde{x}_{t-1}^{(i)}, y_t)\,p(y_t\,|\,y_{0:t-1})}.$$

The right term of the right hand side of this equation is referred to as the ***incremental weight***. The scaling factor $p(y_t\,|\,y_{0:t-1})$ does not depend on the states and does not need to be computed. The sample $(\tilde{x}_t^{(1)},\ldots,\tilde{x}_t^{(N)})$ is called the system of ***particles*** at time t, and $\tilde{x}_{0:t}^{(i)}$ for some i the ***trajectory***, ***history*** or ***path*** of the particle i.

For each time t, one has the set of (weighted) particles $(\tilde{x}_t^{(i)}, \omega_t^{(i)})_{i=1,\ldots,N}$, which is thus moved through the state space if the time t increases. The ***Sequential Importance Sampling (SIS)*** algorithm introduced by Handschin and Mayne [1969] and Handschin [1970] is summarized as follows (Cappé et al. [2007]):

- **Initialization:**

  For $i = 1,\ldots,N$:

  Draw $\quad \tilde{x}_0^{(i)} \sim q_0(x_0)$.

  Assign weights $\quad \tilde{\omega}_0^{(i)} = \dfrac{f_0(\tilde{x}_0^{(i)})}{q_0(\tilde{x}_0^{(i)})}$.

- **Iteration:**

  For $t = 1,\ldots,T$:

For $i = 1, \ldots, N$:

Propagate particle $\quad \tilde{x}_t^{(i)} \sim q_t(x_t \,|\, \tilde{x}_{t-1}^{(i)}, y_t)$.

Compute weight $\quad \tilde{\omega}_t^{(i)} \propto \tilde{\omega}_{t-1}^{(i)} \times \dfrac{f_t(\tilde{x}_t^{(i)} \,|\, \tilde{x}_{t-1}^{(i)}) g_t(y_t \,|\, \tilde{x}_t^{(i)})}{q_t(\tilde{x}_t^{(i)} \,|\, \tilde{x}_{t-1}^{(i)}, y_t)}$.

The paths $\tilde{x}_{0:t}^{(i)}$, $i = 1, \ldots, N$, are independent and identically distributed. But the SIS algorithm is generally a poor algorithm: The weights will degenerate after a few time steps, in the sense that only a few of them will contain nearly all of the probability mass. The weights of most of the particles are near zero and contribute nearly nothing to the empirical distribution and hence to the estimates based on it. This phenomenon is called ***weight degeneracy***.

Sequential Importance Sampling with Replacement (SISR) To avoid this degeneracy, one has to introduce a resampling (selection) step as in the SIR algorithm (see section 3.5.1). The selection step is necessary to avoid degeneracy but in turn increases the variance of the estimates.

There are several possibilities where and when to perform the selection step. The first efficient SMC algorithm, the bootstrap filter of Gordon et al. [1993], performs the selection step after the importance sampling at each iteration, and furthermore uses as proposal distribution q_t the state transition density $f_t(x_t \,|\, x_{t-1})$ (see below). A compromise between these two extremes of not using resampling at all (as in SIS) and using it at each iteration (as in the bootstrap filter) is to resample only then when the weights are going to degenerate. This leads to the following algorithm, called ***Particle Filter*** or ***Sequential Importance Sampling with Replacement (SISR)*** (Cappé et al. [2007]):

- **Initialization:**

  For $i = 1, \ldots, N$:

  Draw $\quad \tilde{x}_0^{(i)} \sim q_0(x_0)$.

  Compute the weights $\quad \tilde{\omega}_0^{(i)} = \dfrac{f_0(\tilde{x}_0^{(i)})}{q_0(\tilde{x}_0^{(i)})}$.

- **Iteration:**

  For $t = 1, \ldots, T$:

  Importance Sampling:

  Für $i = 1, \ldots, N$:

  Propagate particle $\quad \tilde{x}_t^{(i)} \sim q_t(x_t \,|\, \tilde{x}_{t-1}^{(i)}, y_t)$.

  Compute weight $\quad \tilde{\omega}_t^{(i)} \propto \tilde{\omega}_{t-1}^{(i)} \times \dfrac{f_t(\tilde{x}_t^{(i)} \,|\, \tilde{x}_{t-1}^{(i)}) g_t(y_t \,|\, \tilde{x}_t^{(i)})}{q_t(\tilde{x}_t^{(i)} \,|\, \tilde{x}_{t-1}^{(i)}, y_t)}$.

  Selection:

If the weights are degenerate:

Select N particles $(x_t^{(i)})_{i=1,\dots,N}$ from $(\tilde{x}_t^{(i)})_{i=1,\dots,N}$ (with replacement) according to the normalized weights

$$\omega^{(i)} := \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}}.$$

For $i = 1, \dots, N$:

Assign $\tilde{x}^{(i)} \leftarrow x^{(i)}$.

Set the weights $\tilde{\omega}_t^{(i)} \leftarrow 1/N$.

Theoretical investigations on how the empirical distribution

$$\sum_{i=1}^N \omega_t^{(i)} \tilde{x}_{0:t}^{(i)}$$

obtained by this algorithm approximates the joint smoothing density $p(x_{0:1} \,|\, y_{1:t})$ is investigated in Del Moral [1998], Crisan and Doucet [2002], Chopin [2004], Del Moral [2004], Künsch [2005], and Cappé et al. [2005] (see Cappé et al. [2007]).

Effective sample size A known heuristic criterion introduced by Kong et al. [1994] and Liu [1996] which tries to measure the degeneracy of the particles is the ***Effective Sample Size (ESS)***, given as

$$N_{\text{eff}} := \frac{N}{1 + \text{Var}^{p(\cdot|y_{1:t})}(\omega_t(x_{0:t}))} = \frac{N}{\mathbf{E}^{p(\cdot|y_{1:t})}[(\omega_t(x_{0:t}))^2]} \leq N$$

where

$$\omega_t^{(i)} := \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}}$$

are the normalized weights (cf. Doucet et al. [2000]). This cannot be evaluated exactly, but an estimate is given by

$$\hat{N}_{\text{eff}} := \frac{1}{\sum_{i=1}^N (\omega_t^{(i)})^2}.$$

The selection step is only then performed, when the ESS falls below a certain threshold N_{thresh},

$$\hat{N}_{\text{eff}} < N_{\text{thresh}}$$

where the threshold has to be chosen in advance, e.g. $N_{\text{thresh}} = N/2$ or $N_{\text{thresh}} = 2N/3$.

Problems with resampling: Ties With resampling, the sample paths $(x_{0:t}^{(i)})$, $i = 1,\dots,N$, are not any more independent. Even worse: after a few resampling steps, the number of different supporting points $x_s^{(i)}$ for $s < t$ decreases rapidly. Points in time occur where sample paths are tied together, until eventually for $s \ll t$ all $x_t^{(i)}$ have the same ancestor. These ***sample ties*** are a sever problem. One therefore in practice often works only with the filtering distributions, i.e. only with the samples $x_t^{(i)}$ and discards all samples $x_s^{(i)}$ for $s < t$. The same problem occurs with static but unknown states, or parameters θ in the Bayesian setting: In this last case, we have the prior distribution at time $t = 0$, $\theta \sim \pi(\theta)$, and the trivial transition

$$\theta_{t+1} := \theta_t.$$

The initial parameter samples $\theta_0^{(i)}$ obtained by sampling from the prior will then die out after a few resampling steps, until only one of them survives. This is a sever degeneration. In both cases, smoothing and parameter estimation, one has to sample from a discrete distribution. The theoretical convergence results for SMC methods do not hold for discrete distributions. There are several proposals to solve these difficulties, but the problem is still not yet settled. We will return to these issues later.

Practical implementation For the implementation in a computer programme, one should take care with the numerical implementation of the weights: they can be very large and very low, thus causing over- and underflows. Thus typically weights are stored on the log-scale and and updated by addition (see e.g. Cappé et al. [2007]). The normalization step should be done with two steps: One first subtracts the largest log-weight value from all log-weights and then uses normalization on the adjusted log-weights after exponentiating them.

Choice of the proposal distribution

The quality of the algorithm depends crucially on the choice of the proposal distribution q_t.

Optimal proposal The optimal choice is given with the distribution

$$q_t^{\text{opt}}(x_t \,|\, x_{t-1}, y_t) = p(x_t \,|\, x_{t-1}, y_t),$$

see e.g. Doucet et al. [2000]. It is optimal in the sense that it leads to the smallest variance of the importance weights. The problem is that this distribution is usually not available. It can be obtained in the following case (see e.g. Doucet et al. [2000]):

Example:

Let

$$\begin{aligned} x_t &= f(x_{t-1}) + v_{t-1}, \qquad v_{t-1} \sim \mathcal{N}_n(0, \Sigma_v) \\ y_t &= Gx_t + w_t, \qquad w_t \sim \mathcal{N}_n(0, \Sigma_w) \end{aligned}$$

with Gaussian noises v_t and w_t and arbitrary nonlinear conditional mean $f(x_t)$. Defining

$$\Sigma := \left(\Sigma_v^{-1} + G^\top \Sigma_w^{-1} G\right)^{-1}$$
$$m_t := \Sigma\left(\Sigma_v^{-1} f(x_{t-1}) + G^\top \Sigma_w^{-1} y_t\right)$$

one obtains

$$p(x_t \,|\, x_{t-1}, y_t) = \mathcal{N}(m_t, \Sigma)$$

and the importance weigths are given by

$$p(y_t \,|\, x_{t-1}) \propto \exp\left(-\frac{1}{2}\left(y_t - Gf(x_{t-1})\right)^\top \left(\Sigma_v - G\Sigma_w G^\top\right)^{-1}\left(y_t - Gf(x_{t-1})\right)\right).$$

Generally, there are proposals to approximate the optimal importance distribution with Monte Carlo methods, either with importance sampling (Doucet [1997], Doucet [1998]) or with MCMC methods (Berzuini et al. [1997], Liu and Chen [1998]). But both methods are computationally expensive, and there is a lack of theoretical convergence results. Therefore, one has to consider suboptimal proposal functions.

Proposals obtained by linearization Doucet et al. [2000] propose to obtain proposal distributions by linearizations. For the model

$$x_t = f(x_{t-1}) + v_{t-1}, \qquad v_{t-1} \sim \mathcal{N}_n(0, \Sigma_v)$$
$$y_t = g(x_t) + w_t, \qquad w_t \sim \mathcal{N}_n(0, \Sigma_w)$$

they consider two possibilities:

- Local linearization of the state space model: One linearizes the observation distribution in the same way as in the case of the Extended Kalman Filter and obtains similar to the previous example, replacing G by the derivative

$$\tilde{G} = \frac{\partial g(x_t)}{\partial x_t}\Big|_{x_t = f(x_{t-1})},$$

  and replacing y_t by $y_t - g(f(x_{t-1})) + \tilde{G} f(x_{t-1})$:

$$\Sigma_k := \left(\Sigma_v^{-1} + \tilde{G}^\top \Sigma_w^{-1} \tilde{G}\right)^{-1}$$
$$m_t := \Sigma_k\left(\Sigma_v^{-1} f(x_{t-1}) + \tilde{G}^\top \Sigma_w^{-1}\left[y_t - g(f(x_{t-1}) + \tilde{G} f(x_{t-1}))\right]\right).$$

  The proposal function is computed as in the previous example (replacing Σ by Σ_k).

- Local linearization of the optimal importance densities: If $l(x_t) := \log p(x_t \,|\, x_{t-1}, y_t)$ is twice differentiable,

$$l'(x) := \frac{\partial l(x)}{\partial x} \qquad \text{and} \qquad l''(x) := \frac{\partial^2 l(x)}{\partial x \partial x^\top}$$

being the gradient and the Hesse matrix at an arbitrary point x, respectively, and if we additionally assume that $l''(x)$ is postitive definite (which is the case if $l(x)$ is concave), one may define

$$\Sigma(x) := -l''(x)^{-1}$$
$$m(x) := \Sigma(x) l'(x)$$

and use the Taylor expansion of $l(x)$ in x, yielding

$$\begin{aligned} l(x_k) &\approx l(x) + [l'(x)]^\top (x_t - x) + \frac{1}{2}(x_t - x)^\top l''(x)(x_t - x) \\ &= l(x) - \frac{1}{2}(x_t - x - m(x))^\top \Sigma^{-1}(x)(x_t - x - m(x)) + \text{const.} \end{aligned}$$

This suggests using

$$q(x_t \,|\, x_{t-1}, y) = \mathcal{N}(m(x) + x, \Sigma(x))$$

as proposal function. If $p(x_t \,|\, x_{t-1}, y)$ is uni-modal, it is judicious to chose x as the unique mode, which leads to $m(x) = 0$ (Doucet et al. [2000]).

Transition density as proposal A simple choice is to use the transition density

$$q_t(x_t \,|\, x_{t-1}, y_t) = f_t(x_t \,|\, x_{t-1}).$$

The importance weights are in this case porportional to the likelihood function of the observations:

$$\tilde{\omega}_t^{(i)} = g_t(y_t \,|\, x_t).$$

We get the ***bootstrap filter*** of Gordon et al. [1993] (see Cappé et al. [2007]):

- **Initialization:**

  For $i = 1, \ldots, N$:

  Draw $\tilde{x}_0^{(i)} \sim f_0(x_0)$.

  Set the weights $\tilde{\omega}_0^{(i)} = 1/N$.

- **Iteration:**

  For $t = 1, \ldots, T$:

  Importance Sampling:

  For $i = 1, \ldots, N$:

  Propagate particle $\tilde{x}_t^{(i)} \sim f_t(x_t \,|\, \tilde{x}_{t-1}^{(i)})$.

  Compute weight $\tilde{\omega}_t^{(i)} \propto g_t(y_t \,|\, \tilde{x}_t^{(i)})$.

  Selection:

Select N particles $(x_t^{(i)})_{i=1,\dots,N}$ from $(\tilde{x}_t^{(i)})_{i=1,\dots,N}$ (with replacement) according to the normalized weights

$$\omega^{(i)} := \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}}.$$

For $i = 1,\dots,N$:

Assign $\tilde{x}^{(i)} \leftarrow x^{(i)}$.

Set the weights $\tilde{\omega}_t^{(i)} \leftarrow 1/N$.

In general the choice of the transition density as proposal density is not good, because it does not use any information about the observations.

Fixed importance function The simplest choice is the use of a fixed importance function

$$q_t(x_t \,|\, x_{t-1}, y_t) = q(x_t),$$

neither depending on the trajectories nor on the observations. This leads to a rather poor performance of the algorithm.

Auxiliary particle filter (APF)

A variant of the SISR algorithm is obtained if one tries to select the particles in such a way as to favour particles which are more likely to survive in the *next* time step, thus looking more ahead as in the selection step of the usual particle filter (cf. Cappé et al. [2007]). This idea was first used by Pitt and Shephard [1999], and their algorithm was based on auxiliary variables, hence the name Auxiliary Particle Filter for their approach. The presentation here is based on Cappé et al. [2007] which avoids the use of auxiliary variables.

One now considers the joint proposal distribution for the entire path of the new particles $x_{0:t}^{(i)}$ which splits as before,

$$q_{0:t}(dx_{0:t} \,|\, y_{1:t}) := q_{0:t-1}(dx_{0:t-1} \,|\, y_{1:t-1}) q_t(dx_t \,|\, x_{t-1}, y_t),$$

but where the marginal proposal $q_{0:t-1}(dx_{0:t-1} \,|\, y_{1:t-1})$ is given as

$$q_{0:t-1}(dx_{0:t-1} \,|\, y_{1:t-1}) \left(\sum_{i=1}^N v_{t-1}^{(i)} \delta_{x_{0:t-1}^{(i)}}(dx_{0:t-1}) \right)$$

with $\sum_{i=1}^N v_{t-1}^{(i)} = 1$ and $v_{i-1}^{(i)} > 0$. The marginal proposal is now constructed to depend explicitly on the observations up to time t to allow adaptation of the proposal to the observation y_t. This part of the proposal is a discrete distribution with support being the old particle paths $(x_{0:t-1}^{(i)})$, but now with probability mass $v_{t-1}^{(i)}$. This weights may be data dependent with the aim to preselect particles that are a good fit to the new data point y_t. Pitt and Shephard [1999]

suggest to take a point estimate $\mu_t^{(i)}$ of the state, say the mode or mean of $f_t(x_t \mid x_{t-1}^{(i)})$, and computing the weights as the likelihood evaluated at this point:

$$v_{t-1}^{(i)} \propto g_t(y_t \mid \mu_t^{(i)}),$$

or, if the particles are weighted, to choose

$$v_{t-1}^{(i)} \propto \omega_{t-1}^{(i)} g_t(y_t \mid \mu_t^{(i)}).$$

Using this proposal mechanism, the importance ratio between the joint smoothing distribution and the full path proposal q is given by

$$\tilde{\omega}_t^{(i)} \propto \frac{\omega_{t-1}^{(i)}}{v_{t-1}^{(i)}} \times \frac{f_t(\tilde{x}_t^{(i)} \mid \tilde{x}_{t-1}^{(i)}) g_t(y_t \mid \tilde{x}_t^{(i)})}{q_t(\tilde{x}_t^{(i)} \mid \tilde{x}_{t-1}^{(i)}, y_t) p(y_t \mid y_{0:t-1})}$$

for $i = 1, \ldots, N$. The ratio $\omega_{t-1}^{(i)} / v_{t-1}^{(i)}$ is known as the ***first stage weight***. In the original algorithm, there was an additional resampling step. However this is unnecessary and increases the Monte Carlo variance. The ***Auxiliary Particle Filter (APF)*** is then given as follows (Cappé et al. [2007]):

- **Initialization:**

  For $i = 1, \ldots, N$:

  Draw $\tilde{x}_0^{(i)} \sim f_0(x_0)$.

  Compute the weights $\tilde{\omega}_0^{(i)} = \dfrac{f_0(\tilde{x}_0^{(i)})}{q_0(\tilde{x}_0^{(i)})}$.

  Normalize weights:

  For $i = 1, \ldots, N$:

  $$\omega_0^{(i)} := \tilde{\omega}_0^{(i)} / \sum_{j=1}^{N} \tilde{\omega}_0^{(j)}.$$

- **Iteration:**

  For $t = 1, \ldots, T$:

  Selection:

  Select N particle indices $j_i \in \{1, \ldots, N\}$ according to weights $(v_{t-1}^{(i)})_{1 \le i \le N}$

  For $i = 1, \ldots, N$:

  Assign $x^{(i)} \leftarrow \tilde{x}^{(j_i)}$.

  Set the first stage weights $u_{t-1}^{(i)} := \omega_{t-1}^{(j_i)} / v_{t-1}^{(j_i)}$.

  Importance Sampling:

  For $i = 1, \ldots, N$:

Propagate particle $\quad \tilde{x}_t^{(i)} \sim q_t(x_t \,|\, \tilde{x}_{t-1}^{(i)}, y_t)$.

Compute weight $\quad \tilde{\omega}_t^{(i)} \propto u_{t-1}^{(i)} \times \dfrac{f_t(\tilde{x}_t^{(i)} \,|\, \tilde{x}_{t-1}^{(i)}) g_t(y_t \,|\, \tilde{x}_t^{(i)})}{q_t(\tilde{x}_t^{(i)} \,|\, \tilde{x}_{t-1}^{(i)}, y_t) p(y_t \,|\, y_{0:t-1})}$.

Normalize weights:

For $i = 1, \ldots, N$:

$$\omega_t^{(i)} := \tilde{\omega}_t^{(i)} / \sum_{j=1}^{N} \tilde{\omega}_t^{(j)}.$$

Convergence results of the APF and a comparison to SISR (which is not always in favour of the APF) can be found in Johansen and Doucet [2007].

Usage of model structure via Rao-Blackwellisation

Following Doucet et al. [2000], we assume now that the states x_t can be decomposed into (x_t^1, x_t^2) such that the component x_2 can be marginalized out analytically, and such that for any function h the expectation $\mathbf{E}[h]$ can be written in the marginalized form

$$\bar{h} := \mathbf{E}[h] = \frac{\int J(x_{0:t}^1) p(x_{0:t}^1) dx_{0:t}^1}{\int [p(y_{1:t} \,|\, x_{0:t}^1, x_{0:t}^2) p(x_{0:t}^2 \,|\, x_{0:t}^1) dx_{0:t-1}^2] p(x_{0:t}^1) dx_{0:t}^1} = \frac{\int J(x_{0:t}^1) p(x_{0:t}^1) dx_{0:t}^1}{\int p(y_{1:t} \,|\, x_{0:t}^1) p(x_{0:t}^1) dx_{0:t}^1}$$

with

$$J(x_{0:t}^1) := \int h(x_{0:t}^1, x_{0:t}^2) p(y_{1:t} \,|\, x_{0:t}^1, x_{0:t}^2) p(x_{0:t}^2 \,|\, x_{0:t}^1) dx_{0:t}^2.$$

Under the assumption that conditional upon a realization of $x_{0:t}^1$, both $J(x_{0:t}^1)$ and $p(y_{1:t} \,|\, x_{0:t}^1)$ can be evaluated analytically, an estimate $\hat{\bar{h}}$ of $\bar{h} := \mathbf{E}[h]$ can be computed as follows:

$$\hat{\bar{h}} := \frac{\sum_{i=1}^{N} J(x_{0:t}^{1,(i)}) \tilde{\omega}(x_{0:t}^{1,(i)})}{\sum_{j=1}^{N} \tilde{\omega}(x_{0:t}^{1,(i)})}$$

with

$$\tilde{\omega}(x_{0:t}^{1,(i)}) := \frac{p(x_{0:t}^{1,(i)} \,|\, y_{1:t})}{q(x_{0:t}^{1,(i)} \,|\, y_{1:t})}.$$

This technique is called ***Rao-Blackwellisation*** (see Casella and Robert [1996] for a general discussion). Via variance decomposition, one can show that the variances of the importance weights obtained by Rao-Blackwellisation are smaller than those obtained using the direct Monte Carlo methods, see e.g. McEachern et al. [1999]. An example is given with partial Gaussian models (see e.g. Doucet et al. [2000]):

Example: Consider the model given by

$$\begin{aligned} x_t^1 &\sim p(x_t^1 \,|\, x_{t-1}^1) \\ x_t^2 &= A_t(x_t^1) x_{t-1}^2 + B_t(x_t^1) v_t, \qquad v_t \sim \mathcal{N}(0, \mathbf{I}) \\ y_t^2 &= C_t(x_t^1) x_t^2 + D_t(x_t^1) w_t, \qquad w_t \sim \mathcal{N}(0, \mathbf{I}) \end{aligned}$$

(with suitable identity matrices **I**) then, conditional on x_t^1, the model based on states x_t^2 is a linear Gaussian state space model, and the Rao-Blackwellisation method can be realized using the Kalman filter.

Similar is the case if conditioned on x_t^1, the model with states x_t^2 is finite. Rao-Blackwellisation is then done using the discrete filters.

Particle filters with rejection sampling

Instead of using importance sampling in the SMC methods, one could use the accept-reject method (Künsch [2005]). A straightforward approach is: Propose from

$$\frac{1}{N}\sum_{i=1}^{N} f_t(x_t \,|\, x_{t-1}^{(i)})$$

and accept with probability

$$\frac{g_t(y_t \,|\, x_t)}{\sup_x g_t(y_t \,|\, x)}.$$

SMC in combination with MCMC

It has been suggested to combine SMC methods with MCMC, e.g. by applying some Gibbs or Metropolis-Hastings transitions to the particles before or after the selection step, especially to avoid the problems encountered with smoothing and fixed parameter estimation, see e.g. Gilks and Berzuini [2001], Berzuini and Gilks [2001], McEachern et al. [1999], Godsill and Clapp [2001], Fearnhead [2002] and Khan et al. [2005]. This at the first sight appealing idea is computationally expensive and theoretically not founded.

Approximate likelihood

We follow again Cappé et al. [2007] and Doucet et al. [2000]. An approximate likelihood can be abtained from the decomposition

$$p(y_{1:t}) = \prod_{s=1}^{t} p(y_s \,|\, y_{1:s-1})$$

through

$$p(y_s \,|\, y_{1:s-1}) \approx \sum_{i=1}^{N} g_s(y_s \,|\, \tilde{x}_s^{(i)})\omega_s^{(i)}$$

where the samples $(\tilde{x}_t^{(i)})$ are obtained using a one-step ahead prediction based on the filter particles (this is the propagation step in the particle filter), and the ω_s are the normalized importance weights.

Smoother approximation

We already mentioned the degeneracy problem of the smoothing densities caused by the re-sampling step. There are mainly two approaches for improving the smoother samples:

- Additional MCMC steps, and
- Recomputation of weights by backward filtering.

In the case of additional MCMC steps, the complexity is quadratic in the number of time steps. As mentioned, SMC methods with MCMC transitions are also problematic from a theoretical point of view.

One of the proposed backward formulas is the following (see Doucet et al. [2000]): Let $(\tilde{x}_{0:t}^{(i)}, \tilde{\omega}_{0:t}^{(i)})$ be the particles obtained from forward filtering. We then compute new importance weights $\tilde{\omega}_{s|t}^{(i)}$ for the approximation of the smoothing distribution $p(x_s \,|\, y_{1:t})$ as follows:

Initialization at time $s = t$:

- For $i = 1, \ldots, N$: Set $\tilde{\omega}_{t|t}^{(i)} := \tilde{\omega}_t^{(i)}$.

For $s = t-1, \ldots, 0$:

- For $i = 1, \ldots, N$: Set

$$\tilde{\omega}_{s|t}^{(i)} := \sum_{j=1}^{N} \tilde{\omega}_{s+1|t}^{(j)} \frac{\tilde{\omega}_s^{(i)} p(\tilde{x}_{s+1}^{(j)} \,|\, \tilde{x}_s^{(i)})}{\sum_{l=1}^{N} \tilde{\omega}_s^{(l)} p(\tilde{x}_{s+1}^{(j)} \,|\, \tilde{x}_s^{(l)})}.$$

The empirical density obtained from the particles $(\tilde{x}_t^{(i)}, \tilde{\omega}_{s|t}^{(i)})$ approximates then the marginal smoothing density $p(x_s \,|\, y_{1:t})$. It should be noted that the support points $\tilde{x}_t^{(i)}$ remain unchanged, such that the degeneracy problem is not really solved. If the algorithm is running until time T, the memory requirement is $O(TN)$ and the computational complexity is $O(TN^2)$ which is quite sever when the number N of particles is high.

Parameter estimation

Maximum likelihood estimation For an approximate ML estimator, the likelihood at many values θ is needed. Running independent particle filters for many θ's is computationally demanding and leads to a non-smooth likelihood. If one wants to use the EM algorithm, one could try to compute a smooth approximation of the log-likelihood, or to use a stochastic version of the EM algorithm, see Künsch [2001].

Nevertheless, in the sequential setting, computing a new ML estimator $\hat{\theta}$ each time a new observation is available, is not feasible. Needed is an update formula for $\hat{\theta}_{t+1}$ given $\hat{\theta}_t$ and y_{t+1} (and some other statistics). For this recursive estimation almost all proposals rely not only on the filtering densities $p_{t\,|\,1:t}$ but also on the derivative of $p_{t\,|\,1:t}$ with respect to θ, the ***tangent filter***, see e.g. Cérou et al. [2000] and Doucet and Tadić [2003].

Bayesian estimation The simplest approach for Bayesian estimation is to include the parameter θ into the states, with trivial evolution:

$$\theta \equiv \text{const.} \qquad \text{i.e.} \qquad \theta_{t+1} = \theta_t.$$

As already mentioned, the problem is the sample depletion: $(\theta_{t+1}^{(i)})$ is a subsample of $(\theta_t^{(i)})$, and thus after a few resampling steps, only one value of the original sample survives. Possible solutions are:

- Use additional MCMC steps.
- Introduce jittering by adding some noise with small variance to the $\theta_{t+1}^{(i)}$'s. To compensate for the added variance, the $\theta_{t+1}^{(i)}$'s should be shrunk towards their means. The choice of the spread of jitter is difficult. It should decrease for consistent estimation. There is also a principal problem with this approach because the original model is changed.

4 Signal processing, representation and approximation: Wavelets

The concept of the decomposition of a complex system into simpler systems, called atoms, has already been mentioned in earlier chapters: the approximation of a global nonlinear differentiable model by local linear models with local model networks, or the continuous superposition of hysterons for building the Preisach hysteresis are examples of such atomic decompositions. There is an obvious difference: Whereas in the case of the Preisach model, the superposition of hysterons gives the "true" model, the case of local model networks is usually seen as approximation to the true model. But, as is usually assumed, the approximation to the true model is the better the more local models we include into the local model network. We may formulate the difference in the following way: whereas the Preisach hysteresis is *represented* by the atomic decomposition into hysterons, the nonlinear model is *approximated* by the local model network. Of course, the two notions are strongly connected: If we pick out N points on the Preisach plane (α_k, β_k) and choose appropriate Dirac distributions $w_k\delta_k := w_k\delta_{(\alpha_k,\beta_k)}$, $k = 1,\ldots,N$, such that

$$\mu \approx \sum_{k=1}^{N} w_k \delta_k,$$

we have the *approximation*

$$\Gamma^{\mu} = \int \Gamma^{\alpha,\beta} d\mu \approx \sum_{k=1}^{N} w_k \Gamma^{\alpha_k,\beta_k},$$

and, vice versa, if in the case of local model networks we let the number of local models go to infinity, we may get a *representation*

$$\sum_{k=1}^{\infty} w_k \Gamma_k$$

of the true nonlinear model. Thus, representations lead to approximations and vice versa.

To make things clear: In the case of the local model networks, we indeed use the decomposition for approximation, at least indirectly, see chapter 1. In contrast, in the case of the Preisach model, we did *not* use the decomposition for the approximation given in chapter 2 which was the basis for the variant of the LOLIMOT algorithm we presented there (nevertheless, the lookup-table approach mentioned in section 2.2.3 does exactly this, with an additional linear interpolation step). Instead, we used a *further* decomposition of the primitive function F which in turn is derived from the Preisach measure μ. Note that μ is the weight in the first decomposition.

Returning to the local model networks, we do not directly approximate the differentiable global model by linear local models. We actually approximate the nonlinear output map

$$\eta(x(t))$$

where

$$x(t) = \big(u(t), u(t-1), \ldots, u(t-n_u), y(t-1), \ldots, y(t-n_y)\big)^\top$$

in the case of NARX models or

$$x(t) = \big(u(t), u(t-1), \ldots, u(t-n_u), \hat{y}(t-1), \ldots, \hat{y}(t-n_y)\big)^\top$$

in the case of NOE models (see chapter 1).

Thus, in both cases, Preisach hysteresis and local model networks, we found that the description of the systems can be reduced to the description of some characteristic function which has to be approximated: In the case of the Preisach hysteresis, this characteristic function is the primitive function F, and in the case of local model networks, it is the output function η. We need a representation (or approximation) for (multi-dimensional) real functions. The original local model networks (and the LOLIMOT algorithm) use superpositions of linear functions weighted by normalized Gaussian functions. We proposed to replace the Gaussian weight functions by decision tree based weight functions to get more flexibility (see chapter 1). In chapter 5 we want to replace the product of weight functions and linear functions by wavelet (or wavelet packet) bases and their corresponding coefficients.

The theory which investigates the possibilities and properties of function approximation is (Constructive) Approximation Theory. The aim is the approximation of a complicated function by simpler functions, called approximants. If these approximants come from a linear space, the resulting approximation is called linear approximation, otherwise we are concerned with nonlinear approximation. One should not misunderstand the term “nonlinear”: it has no connection with the same term used in “nonlinear system”, nor does “linear approximation” mean that the approximated function or the approximants itself are linear functions. In both cases, linear and nonlinear approximation, the approximated function as well as the approximants are generally nonlinear as functions. Only the set of all approximants (of some given approximation level) forms a linear space in the case of linear approximation. In the case of nonlinear approximation, they do not, i.e. the sum of two approximants of a given level does not need to be an approximant of the same level.

An important case of nonlinear approximation is n-term approximation: Choose the approximants as a sum of n terms such that approximation is best. Candidates are free-knot splines and wavelet bases expansions. We will see that in the case of multi-dimensional functions wavelets are to be preferred.

A step further goes the so-called highly nonlinear approximation. Wavelet packets and approximation from libraries fall in this category. The latter subsumes also neural networks and the approximants consisting of products of normalized Gaussian (or decision tree based) weight functions and linear functions used in the local model networks.

Whereas the n-term approximation with wavelets (or wavelet packets) is well-understood and leads to interesting connections between n-term approximation, thresholding procedures,

smoothness spaces and sparsity properties of the wavelet coefficients, few is known about the highly non-linear approximations. This is one reason why we want to use wavelets.

Another reason is the successful combination of wavelet and Bayes methods: By putting suitable priors on the wavelet coefficients and by applying suitable loss functions, certain wavelet coefficients of a real signal are set to zero, which results in a kind of regularized approximated signal. This procedure has been used for e.g. denoising of images and audio signals. We will use similar ideas for our identification procedure in chapter 5.

Overview The first section is devoted to wavelets in general as well as to the continuous and discrete wavelet transforms, and multiresolution analysis. Nonlinear approximation and the deep connections between approximation, interpolation and smoothness spaces will be presented next, followed by a short overview of highly nonlinear approximation and greedy algorithms. We proceed with procedures for denoising. We then present some existing examples where wavelets are used in nonparametric identification of nonlinear systems. An analogue to the multiresolution analysis for functions in the case of linear systems will be shortly mentioned in the last subsection.

Contributions Like the previous chapter, this chapter also combines topics found in several places in the literature, eventually leading, together with the topics presented in the previous chapters, to our new model in chapter 5.

4.1 Wavelets

4.1.1 Signal analysis

Signals We adopt the "working definition" of Jaffard et al. [2001]: ***Signals*** are sequences of numbers, typically coming from measurements (this includes of course hidden signals which are practically not measurable). One then may think of these signals as functions of time like originating from music or speech, or functions of position. It is equally important to consider two-dimensional signals e.g. stemming from images, where a function value $f(x,y)$ is assigned to each grey-scale value, x and y being the coordinates of the position. Signal processing is then the task to code or transform these signals in a suitable way, where different aims lead to similar techniques: transferring of signals over noisy communication channels, compression of signals to reduce the amount of neccessary memory for storage, restoration of signals which are observed from noisy measurements, or, as is our case, for finding a model for some observed signals. Modelling may thus be seen as a kind of data reduction. In all cases, the signals have to be analyzed and transformed in a way such that important information may be separated from neglectable or even unwanted information.

The right representation Signal analysis thus serves to the following task: to represent a signal in a way as to make "explicit certain entities or types of information" (David Marr, cited in Jaffard et al. [2001]). To reveal these sought informations, representations of signals are often given by atomic decompositions; different atoms are possible, and thus different

representations. One has to choose suitable atoms. These atoms are often given by certain ***test functions*** with which the signal is folded. The test functions show in purety exactly this kind of information one is interested in.

Thus, which analysis is suited for which signals? Jaffard et al. [2001] for example distinguish between three kinds of signals and corresponding test signals. An overview over these analyses for different kinds of signals is shown in table 4.1.

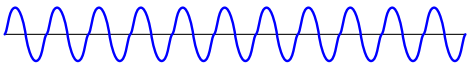
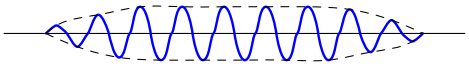
Signals	Analysis	Typical test signal
Stationary	Fourier	
Quasi-stationary	Time-frequency wavelets	
Some transient	Time-scale wavelets	

Table 4.1: Signals and preferred analysis

Phase space and Heisenberg's uncertainties The most known and oldest one of these techniques is surely the Fourier transform. It is best suited for stationary signals and decomposes the signal into a linear combination of (sine and cosine) waves (for definition and properties of the Fourier transform $\mathscr{F}$ see the appendix).

The Fourier transformed function $\hat{f}$ of a function (signal) $f \in L^2(\mathbb{R})$ reveals information on the frequency distribution of f: the value $\hat{f}(\omega)$ corresponds to the amplitude of the frequency ω on the global function f. There is no information on local properties of f. Sometimes one would like to have information on the frequency distribution at each time t, i.e. one would like to assign to f a function $(\mathscr{D}f)(t,\omega)$ for each time t and each frequency ω which reveals information on how much the frequency ω is contained in the signal f at time t. The set of all pairs (t,ω), $t,\omega \in \mathbb{R}$, is called the ***phase space***, and $\mathscr{D}f$ is called the ***phase space representation*** of f (see Louis et al. [1994]). If g_{t_0,ω_0} is a function which is "concentrated" or "localized" at t_0 in the time domain and at ω_0 in the frequency domain (i.e. regarding $\hat{f}$), then such a phase space representation is given by

$$(\mathscr{D}f)(t_0,\omega_0) := \langle g_{t_0,\omega_0}, f\rangle_{L^2(\mathbb{R})}.$$

We define "localization" as follows (following Louis et al. [1994]):

Definition 4.1: *Let $g \in L^2(\mathbb{R})$ with $\|g\|_{L^2(\mathbb{R})} = 1$ and*

$$-\infty < t_0 := \int_{\mathbb{R}} t|g(t)|^2 dt < \infty,$$
$$-\infty < \omega_0 := \int_{\mathbb{R}} \omega|\hat{g}(\omega)|^2 d\omega < \infty.$$

*Then we say that g is **localized at the time** t_0 with uncertainty*

$$\int_{\mathbb{R}} (t-t_0)^2 |g(t)|^2 dt,$$

*g is **localized at the frequency** ω_0 with uncertainty*

$$\int_{\mathbb{R}} (\omega-\omega_0)^2 |\hat{g}(\omega)|^2 d\omega,$$

*and g is **localized at the phase point** (t_0,ω_0) with uncertainty*

$$\int_{\mathbb{R}} (t-t_0)^2 |g(t)|^2 dt \int_{\mathbb{R}} (\omega-\omega_0)^2 |\hat{g}(\omega)|^2 d\omega.$$

Thus, t_0 is the mean value of the function $|g|^2$ and the uncertainty is nothing else but the variance, analogously for ω_0 and $|\hat{g}|^2$.

We would like the function g_{t_0,ω_0} to be localized at the phase point (t_0,ω_0) with uncertainty 0. That this is not achievable is the content of the ***Heisenberg uncertainty principle*** (see again e.g. Louis et al. [1994]):

Theorem 4.1 (Heisenberg uncertainty principle)**:** *Let $g \in L^2(\mathbb{R})$ with $\|g\|_{L^2(\mathbb{R})} = 1$. Then the uncertainty at an arbitrary phase point (t_0,ω_0) is never less than $1/4$, i.e.*

$$\int_{\mathbb{R}} (t-t_0)^2 |g(t)|^2 dt \int_{\mathbb{R}} (\omega-\omega_0)^2 |\hat{g}(\omega)|^2 d\omega \geq 1/4$$

for all $t_0,\omega_0 \in \mathbb{R}$.

The best possible localization in the time domain at t_0 is attained by the Dirac measure $\delta(\cdot - t_0)$, the best localization in the frequency domain at ω_0 is attained by $e^{-i\omega_0 \cdot}$. But neither of them is a phase space localization, because they do not localize in the respective complementary domain. The function with minimal uncertainty at (t_0,ω_0) is

$$g_{t_0,\omega_0}(t) := \pi^{-1/4} e^{-i\omega_0 t} e^{-(t-t_0)^2/2}.$$

Here, the uncertainty is indeed $1/4$. This is the reason why Gabor introduced the windowed Fourier transform.

Time-frequency wavelets The ***windowed Fourier transform*** or ***Gabor transform*** $\mathcal{G}$ introduced by Gabor in 1946 is the first example of a time-frequency wavelet:

$$(\mathcal{G}f)(s,\xi) := \int_{-\infty}^{+\infty} f(t) g_s(t) e^{-it\cdot\xi} dt$$

where

$$g_s(t) := g(t-s)$$

is the ***window*** translated by s. The domain of the transformed signal is two-dimensional: The s-axis denotes time, the ξ-axis denotes frequency. The window is often chosen to be the Gaussian function

$$g(t) := \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{t^2}{2\sigma^2}\right)$$

with fixed $\sigma > 0$. It was expected that every function in $L^2(\mathbb{R})$ can be decomposed with Gabor wavelets as atoms. But the Balian-Low theorem (Balian [1981] and Low [1985]) shows that this is not the case, and that this negative result is valid for any smooth, well-localized function g.

Malvar-Wilson wavelets More recently, Henrique Malvar and Kenneth Wilson discovered time-frequency wavelets with good algorithmic properties, particularly suited for coding speech and music. The Gabor transform uses windows of a fixed size. One main difference to this is that in the case of the Malvar-Wilson wavelets windows with variable sizes are used. Following Jaffard et al. [2001], we begin with an arbitrary partition of the real line into intervals $[a_j, a_{j+1}]$, $j \in \mathbb{Z}$ with

$$\cdots < a_{-2} < a_{-1} < a_0 < a_1 < a_2 < \cdots$$

and

$$\lim_{j\to+\infty} a_j = +\infty, \qquad \lim_{j\to-\infty} a_j = -\infty.$$

If $l_j := a_{j+1} - a_j$ denotes the length of the interval $[a_j, a_{j+1}]$, let $\alpha_j > 0$ be positive numbers small enough such that

$$l_j \geq \alpha_j + \alpha_{j+1} \qquad \text{for all } j \in \mathbb{Z}.$$

The windows w_j are essentially characteristic functions of the intervals $[a_j, a_{j+1}]$ which overlap in the disjoint intervals $(a_j - \alpha_j, a_j + \alpha_j)$. We impose on the w_j's the following conditions:

- $0 \leq w_j(t) \leq 1$ for all $t \in \mathbb{R}$,
- $w_j(t) = 1$ if $a_j + \alpha_j \leq t \leq a_{j+1} - \alpha_{j+1}$,
- $w_j(t) = 0$ if $t \leq a_j - \alpha_j$ or $t \geq a_{j+1} + \alpha_{j+1}$,
- $w_j^2(a_j + \tau) + w_j^2(a_j - \tau) = 1$ if $|\tau| \leq \alpha_j$,
- $w_{j-1}(a_j + \tau) = w_j(a_j - \tau)$ if $|\tau| \leq \alpha_j$.

The windows w_j can be chosen to be infinitely often differentiable, and we have:

$$\sum_{j=-\infty}^{+\infty} w_j^2(t) = 1 \qquad \text{for all } t \in \mathbb{R}.$$

The Malvar-Wilson wavelets appear in two forms. The first is given by

$$u_{j,k} := \sqrt{\frac{2}{l_j}} w_j(t) \cos\left[\frac{\pi}{l_j}\left(k + \frac{1}{2}\right)(t - a_j)\right] \qquad \text{for } j \in \mathbb{Z},\ k \in \mathbb{N}.$$

The second form alternates between cosines and sines according to whether j is even or odd:

$$u_{j,k} := \begin{cases} \sqrt{\frac{2}{l_j}} w_j(t) \cos\left[\frac{k\pi}{l_j}(t-a_j)\right] & \text{for } j \in 2\mathbb{Z}, \quad k = 1,2,\ldots, \\ \sqrt{\frac{2}{l_j}} w_j(t) & \text{for } j \in 2\mathbb{Z}, \quad k = 0, \\ \sqrt{\frac{2}{l_j}} w_j(t) \sin\left[\frac{k\pi}{l_j}(t-a_j)\right] & \text{for } j \in 2\mathbb{Z}+1, \quad k = 1,2,\ldots. \end{cases}$$

In both cases, the functions $u_{j,k}$, $j \in \mathbb{Z}$, $k \in \mathbb{N}$, constitute an orthonormal basis for $L^2(\mathbb{R})$.

Musical notation The Malvar-Wilson wavelets show close similarities to musical notation: musical notation shows duration in time *and* frequency. A translation from a given score into an acoustic signal is relatively easy (we are not talking about *music*!). The opposite way, the construction of time-frequency wavelets only from the given signal is difficult due to the Heisenberg uncertainty principle. The decomposition of a signal in an orthonormal basis of Malvar-Wilson wavelets imitates writing music using a musical score (see figure 4.1). But in contrast to a musical score, there are infinitely many ways to decompose a signal into orthonormal bases of Malvar-Wilson wavelets (see Jaffard et al. [2001]): One first chooses a segmentation of the signal and then uses a traditional Fourier analysis on the delimited pieces. One again has to use prior knowledge, e.g. that tones have a minimal duration in time and occur only in prescribed frequencies (harmonics).

Time – frequency – scale The third kind of signal analysis is based on the time-scale wavelets. Here, the theory is much more complete than in the case of time-frequency wavelets, and fast and easy algorithms exist. In the case of time-scale wavelets, the resolution in frequencies is replaced by the resolution in scales: One looks at a given function (signal, image) f as through different looking glasses and decomposes f successively into a coarse part and finer and finer details. The disadvantage of time-scale wavelets is that they are not able to detect periodicity, e.g. frequencies if f is an acoustic signal, or patterns if f is an image. To overcome this deficiency, the wavelet packets have been introduced. These can be seen as an alternative to Malvar-Wilson wavelets. We will describe time-scale wavelets and and wavelet packets in more detail in the following sections. In the sequel, we will often drop the term "time-scale" and speak simply of "wavelets".

4.1.2 Time-scale wavelets

History of time-scale wavelets We follow here Jaffard et al. [2001] and DeVore [1998]. The first orthonormal (time-scale) wavelet basis is the Haar basis, constructed by A. Haar in 1910. The Haar wavelet for the Hilbert space $L^2(\mathbb{R})$ is defined as

$$H(x) := \begin{cases} 1 & \text{if } x \in [0,1/2), \\ -1 & \text{if } x \in [1/2,1), \\ 0 & \text{else,} \end{cases}$$

and the Haar Basis is obtained through translations and dilations of this function H. Another basis was constructed by Walsh in 1923, which now can be seen as wavelet packet basis. The

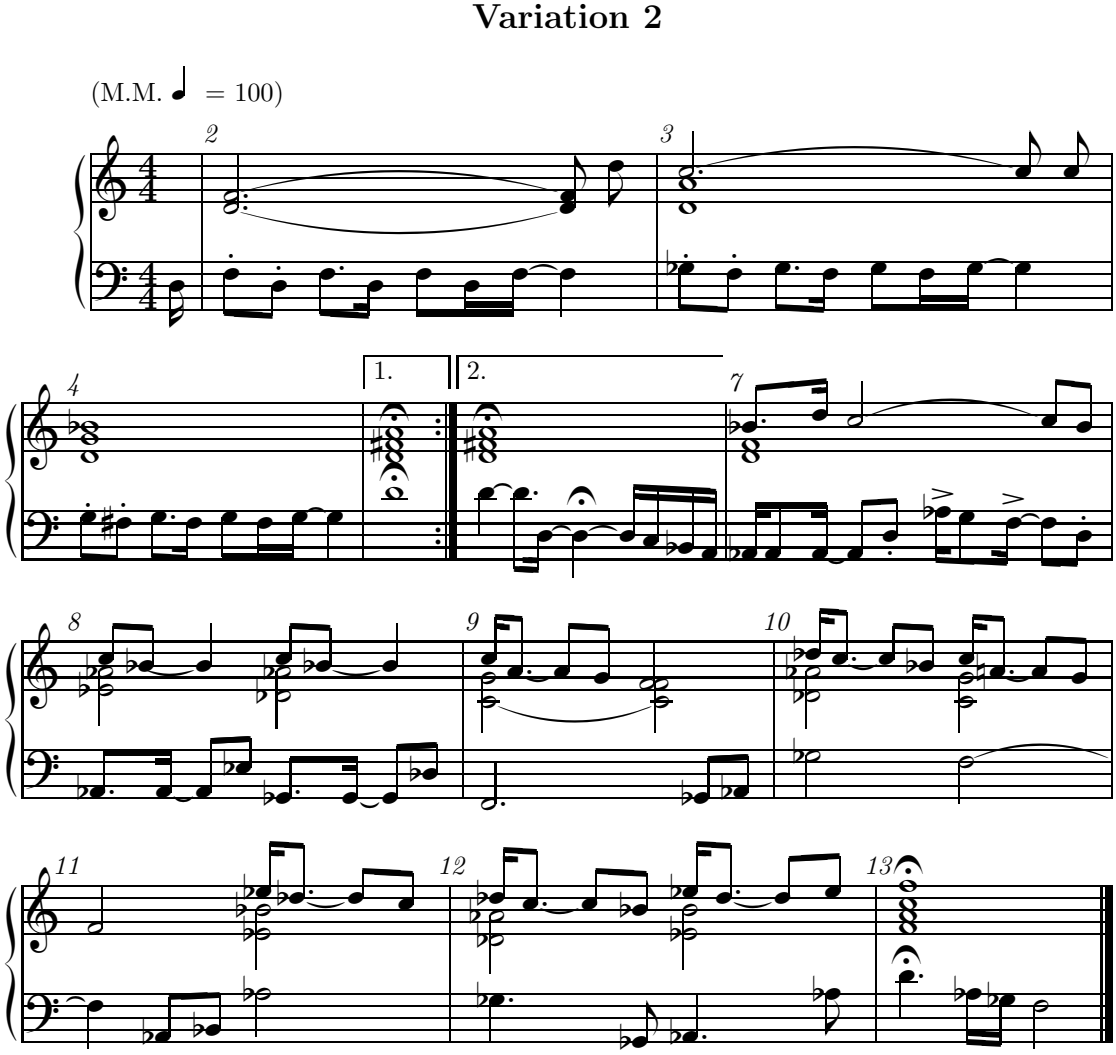


Figure 4.1: An example of time-frequency wavelets: Time is on the *x*-axis, frequency on the *y*-axis (excerpt from: Tupperwaliationen (2006/2007), composed by Anne Fuchs (with permission of the composer; all rights are with her)

development into wavelet theory came then from several disciplines: spline approximation theory, signal and image processing and harmonic analysis. The notion "wavelet" was introduced by J. Morlet and A. Grossmann in 1982. In the year 1985, Yves Meyer constructed wavelet bases with good smoothness properties. He also was very important in the development of the foundations of wavelet theory. During the same time, S. Mallat and Y. Meyer created the multiresolution analysis, an essential tool for the construction of wavelets (see Mallat [1989]). Ancestors to multilevel decomposition were

- multigrid methods in numerical computation,
- box splines in approximation theory, and
- Littlewood-Paley theory in harmonic analysis.

A great impetus came also from the discovery of the first orthogonal wavelet basis with arbitrarily smooth functions of compact support constructed by Ingrid Daubechies in 1987 (see Daubechies [1988]).

Good properties for wavelets The property of the Daubechies wavelets to have compact support is important for the good localization in time of the corresponding bases: Small changes in the signal result in small changes in only a few wavelet coefficients. Apart from the properties of orthogonality and compact support, a third property is of interest: that the wavelets have a sufficient number of vanishing moments. A function $\psi \in L^1(\mathbb{R})$ is said to have m ***vanishing moments*** or to be of ***order*** m if

$$\int_{-\infty}^{\infty} x^k \psi(x) dx = 0 \qquad \text{for all } k = 0, \ldots, m-1,$$

which particularly induces for the derivatives $\hat{\psi}^{(k)} := d^k \hat{\psi} / d\omega^k$ of the Fourier transformed function $\hat{\psi}$, that :

$$\hat{\psi}^{(k)}(0) = 0 \qquad \text{for } k = 0, \ldots, m-1$$

holds. If the wavelet ψ has enough vanishing moments, then the wavelet coefficients are small in these regions where the function to be analysed is smooth. Where the function shows large changes or jumps, the wavelet coefficients are large. We shortly mention that for a function $\psi \in L^1(\mathbb{R})$ with compact support, the following equivalence is valid:

$$\int_{\mathbb{R}} x^k \psi(x) dx = 0 \qquad \text{for all } k \in \mathbb{N}$$

if and only if $\psi \equiv 0$. Thus, wavelets with compact support are of finite order (see e.g. Louis et al. [1994]).

4.1.3 The continuous wavelet transform

The continuous wavelet transform gives a theoretical basis where methods for practical applications like discrete wavelet transform or multiresolution analysis can be built upon. The

continuous wavelet transform establishes operators which transform $L^2(\mathbb{R})$ isometrically (but not surjectively!) into the function space

$$L^2\left(\mathbb{R}_{>0}\times\mathbb{R}, db\frac{da}{a^2}\right).$$

Here, a different measure than the Lebesgue measure is used in the a-variable. The continuous wavelet transform can also be formulated for the d-dimensional case where the functions come from $L^2(\mathbb{R}^d)$, with $d>1$. But we nevertheless restrict ourselves to the one-dimensional case $d=1$. Higher dimensions will be considered later in the more application relevant situations. For this later use, we formulate the following admissibility condition for general dimensions $d\geq 1$. We follow mainly Louis et al. [1994].

Definition 4.2 (Wavelets)**:** *A function $\psi\in L^2(\mathbb{R}^d)$ which fulfills the* ***admissibility condition***

$$0<2\pi\int_{\mathbb{R}}|\hat{\psi}(t\xi)|^2\frac{dt}{|t|}=:C_\psi<+\infty \quad \textit{for almost all } \xi\in\mathbb{R}^d$$

is called a ***(time-scale) wavelet****.*

The Continuous Wavelet Transform (CWT) in dimension $d=1$ is then defined with respect to a given wavelet ψ:

Definition 4.3 (Continuous Wavelet Transform (CWT))**:** *Let $\psi\in L^2(\mathbb{R})$ be a wavelet. The* ***wavelet transformed function*** *$\mathscr{W}_\psi f$ of a function $f\in L^2(\mathbb{R})$ to the wavelet ψ is given by*

$$(\mathscr{W}_\psi f)(a,b)=\frac{1}{\sqrt{C_\psi}}|a|^{-1/2}\int_{\mathbb{R}}f(x)\psi\left(\frac{x-b}{a}\right)dx \quad \textit{for all } a>0,\, b\in\mathbb{R}.$$

The operator

$$\mathscr{W}_\psi : L^2(\mathbb{R})\longrightarrow L^2\left(\mathbb{R}_{>0}\times\mathbb{R}, db\frac{da}{a^2}\right)$$

is called the ***continuous wavelet transform (CWT)*** *with respect to the wavelet ψ.*

It is possible to choose $\|\psi\|=1$ or $C_\psi=1$, but not necessarily both simultaneously. For a given wavelet ψ, the so-called ***mother wavelet***, we define the translated and dilated versions $\psi_{a,b}\in L^2(\mathbb{R})$ of ψ by

$$\psi_{(a,b)}(x):=\frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right),\quad a>0, b\in\mathbb{R}.$$

Then the mapping $(a,b)\mapsto\psi_{(a,b)}$ is continuous from $\mathbb{R}_{>0}\times\mathbb{R}$ to $L^2(\mathbb{R})$. The normalization $1/\sqrt{a}$ is chosen such that $\|\psi_{(a,b)}\|_2=\|\psi\|_2$. If we work in other L^p spaces, different normalizations may be better suited; we return to this topic later. With the translated and dilated wavelets $\psi_{(a,b)}$, the continuous wavelet transform $\mathscr{W}_\psi$ can be written in the following form:

$$(\mathscr{W}_\psi f)(a,b)=\frac{1}{\sqrt{C_\psi}}\langle f,\psi_{(a,b)}\rangle=\frac{1}{\sqrt{C_\psi}}\int_{\mathbb{R}}f(x)\overline{\psi_{(a,b)}(x)}dx \quad \text{for all } a>0,\, b\in\mathbb{R}$$

where the bar denotes the complex conjugate. From $|(\mathscr{W}_\psi f)(a,b)|\leq\|f\|\|\psi\|$ and from the continuity of $(a,b)\mapsto\psi_{(a,b)}$, the continuity of $(\mathscr{W}_\psi f)(a,b)$ on $\mathbb{R}_{>0}\times\mathbb{R}$ follows.

Theorem 4.2: *The wavelet transform with respect to the wavelet ψ*

$$\mathscr{W}_\psi : L^2(\mathbb{R}) \longrightarrow L^2\left(\mathbb{R}_{>0}\times\mathbb{R}, db\frac{da}{a^2}\right)$$

is a partial isometry, i.e.

$$\|\mathscr{W}_\psi f\|_{L^2(\mathbb{R}_{>0}\times\mathbb{R},dbda/a^2)} = \|f\|_{L^2(\mathbb{R})}.$$

Inversion Let now φ be a second wavelet (in $L^2(\mathbb{R}^d)$) where the admissibility condition with C_φ is fulfilled. Let further be the following ***joint admissibility condition*** be fulfilled:

$$0 < 2\pi\int_{\mathbb{R}} \hat{\psi}(t\xi)\overline{\hat{\varphi}(t\xi)}\frac{dt}{|t|} = C_{\psi\varphi} < +\infty \quad \text{for almost all } \xi\in\mathbb{R}^d.$$

We then have (again for $d=1$):

Theorem 4.3: *Let $\psi,\varphi\in L^2(\mathbb{R})$ be wavelets such that the admissibility conditions with C_ψ and C_φ and the joint admissibility condition with $C_{\psi\varphi}$ be fulfilled. Then*

$$C_\psi C_\varphi \langle \mathscr{W}_\psi f, \mathscr{W}_\varphi g\rangle_{L^2(\mathbb{R}_{>0}\times\mathbb{R},dbda/a^2)} = C_{\psi\varphi}\langle f,g\rangle_{L^2(\mathbb{R})} \qquad \textit{for all } f,g\in L^2(\mathbb{R}).$$

Inversion formulas

Theorem 4.4 (Inversion of continuous wavelet transform)**:** *Let $\mathscr{W}_\psi$ be the continuous wavelet transform to the wavelet ψ. Then, the adjoint operator*

$$\mathscr{W}_\psi^* : L^2\left(\mathbb{R}_{>0}\times\mathbb{R}, db\frac{da}{a^2}\right) \longrightarrow L^2(\mathbb{R})$$

where $g\mapsto\mathscr{W}_\psi^* g$ with

$$(\mathscr{W}_\psi^* g)(x) = C_\psi^{-1/2}\int_{\mathbb{R}}\int_{\mathbb{R}} |a|^{-1/2}\psi\left(\frac{x-b}{a}\right) g(a,b)db\frac{da}{a^2}$$

inverts the continuous wavelet transform $\mathscr{W}_\psi$ on its range, i.e.

$$\mathscr{W}_\psi^*\mathscr{W}_\psi = \mathrm{Id} \qquad \textit{and} \qquad \mathscr{W}_\psi\mathscr{W}_\psi^* = P_{\mathscr{W}_\psi(L^2(\mathbb{R}))},$$

where $P_{\mathscr{W}_\psi(L^2(\mathbb{R}))}$ is the orthogonal projection to the range of $\mathscr{W}_\psi$.

4.1.4 The discrete wavelet transform

The continuous wavelet transform serves to the right understanding of wavelets. For practical computations, it is not necessary to know $\mathscr{W}_\psi f$ at all points $(a,b)\in(\mathbb{R}_{>0}\times\mathbb{R})$ for any given function $f\in L^2(\mathbb{R})$. Indeed, there is a large redundancy concerning these values. If one wants to exactly reproduce f from $\mathscr{W}_\psi f$, it is enough to know $\mathscr{W}_\psi f$ on certain discrete points (a,b).

Wavelet frames Following Louis et al. [1994], we define the grid

$$\left\{(a_0^{-j}, kb_0a_0^{-j}) \in \mathbb{R}_{>0} \times \mathbb{R} \;\middle|\; j,k \in \mathbb{Z}\right\}$$

with $a_0 > 1$, $b_0 > 0$, and for a given wavelet ψ the corresponding function set

$$\left\{\psi_{j,k}^{a_0,b_0}(\cdot) := a_0^{j/2}\psi(a_0^j \cdot - kb_0) \;\middle|\; j,k \in \mathbb{Z}\right\}.$$

We then define:

Definition 4.4: *Let $a_0 > 1$, $b_0 > 0$ and $\psi \in L^2(\mathbb{R})$. We say that the function set*

$$\left\{\psi_{j,k}^{a_0,b_0}(\cdot) \;\middle|\; j,k \in \mathbb{Z}\right\}$$

*is a **wavelet frame** for $L^2(\mathbb{R})$, if there exist constants $A,B > 0$ such that*

$$A\|f\|_{L^2(\mathbb{R})}^2 \leq \sum_{j\in\mathbb{Z}}\sum_{k\in\mathbb{Z}} |\langle \psi_{j,k}^{a_0,b_0}, f\rangle_{L^2(\mathbb{R})}|^2 \leq B\|f\|_{L^2(\mathbb{R})}^2.$$

*We say that the triple (ψ, a_0, b_0) **generates** the frame. The constants A and B are called the **bounds of the frame**. The frame is said to be **tight** if $A = B$.*

The question arises under which conditions the triple (ψ, a_0, b_0) generates a wavelet frame. It can be shown that the admissibility condition on the wavelet ψ is necessary (this admissibility condition was not required in the definition). A detailed description of necessary and sufficient conditions on the triple (ψ, a_0, b_0) can be found in Louis et al. [1994]. The usual choice is $a_0 := 2$ and $b_0 := 1$.

To each wavelet frame generated by (ψ, a_0, b_0), we can assign the operator

$$T : L^2(\mathbb{R}) \longrightarrow \ell^2(\mathbb{Z}^2), \quad (Tf)_{j,k} := \langle \psi_{j,k}^{a_0,b_0}, f\rangle_{L^2(\mathbb{R})}.$$

For this operator,

$$A^{1/2}\|f\|_{L^2(\mathbb{R})} \leq \|Tf\|_{\ell^2(\mathbb{Z}^2)} \leq B^{1/2}\|f\|_{L^2(\mathbb{R})}$$

holds, and T is thus continuous with

$$\|T\| \leq B^{1/2},$$

and continuously invertible on its range,

$$\left\|T^{-1}\big|_{T(L^2(\mathbb{R}))}\right\| \leq A^{1/2},$$

i.e. f can be reconstructed from the discrete values

$$(Tf)_{j,k} = \sqrt{C_\psi}(\mathscr{W}_\psi f)(a_0^{-j}, kb_0a_0^{-j}).$$

One just has to determine T^{-1}. We call the operator T the ***Discrete Wavelet Transform (DWT)*** belonging to the triple (ψ, a_0, b_0), and correspondingly T^{-1} the ***Inverse Discrete Wavelet Transform (IDWT)***.

Dyadic frames As mentioned, the usual choice for a_0 and b_0 is $a_0 := 2$ and $b_0 := 1$. This leads to the ***dyadic wavelet frames***:

$$\left\{ \psi_{j,k}^{2,1}(\cdot) := 2^{j/2}\psi(2^j \cdot -k) \;\middle|\; j,k \in \mathbb{Z} \right\}.$$

In this case, we will write shortly

$$\psi_{j,k} := \psi_{j,k}^{2,1}.$$

Example: The Haar wavelets Let

$$H(x) := \begin{cases} 1 & \text{if } x \in [0,1/2), \\ -1 & \text{if } x \in [1/2,1), \\ 0 & \text{else,} \end{cases}$$

be the ***Haar function*** and set $\psi(x) := H(x)$. Then

$$\psi_{j,k}(x) := 2^{j/2}\psi(2^j x - k) \qquad \text{for } j,k \in \mathbb{Z}$$

builds an orthogonal wavelet basis of $L^2(\mathbb{R})$, called the ***Haar basis***. This is the easiest example of a wavelet basis. It can be generalized to the Daubechies wavelet bases, which cannot be given explicitly (except in the case of the Haar wavelet), and which we present after having introduced the important tool of multiresolution analysis.

Construction of biorthogonal wavelets with compact support The wavelets considered until now constitute orthonormal bases in $L^2(\mathbb{R})$. In some situations it is desirable to relax the orthogonality to gain more flexibility in the construction of wavelets. Thus, for example, the Haar wavelet is the only known wavelet that is compactly supported, orthogonal and simultaneously symmetric. But the Haar wavelet is non-smooth (not even continuous) and has only one vanishing moment. While smoother wavelets like higher-order Daubechies wavelets are preferable in applications, they have the disadvantage not to be symmetric. Nevertheless, symmetry often is desirable in applications. Giving up orthogonality, one only requires that the wavelets form a Riesz basis in $L^2(\mathbb{R})$ (see the appendix for definition and properties). In this case, one calls them ***biorthogonal wavelets***. The main difference is that with orthogonal wavelets, direct and inverse transform can be done using the same wavelet; with biorthogonal wavelets, one has to use a second (different) wavelet for the inverse transform.

4.1.5 Multiresolution analysis and Fast Wavelet Transform (FWT)

The discrete wavelet transform is strongly connected with multiresolution analysis. Two operations are important in multiresolution analysis: shift and dilation. This leads first to the shift invariant spaces. We follow here DeVore [1998] and Louis et al. [1994].

Shift invariant spaces If for some dimension $d \geq 1$ an arbitrary function f is defined on $\mathbb{R}^d$, and if further $k \in \mathbb{Z}^d$ and $a > 0$ a real number are given, then call

- $f(\cdot - k)$ the (integer) ***shift*** of f by k,
- $f(a\cdot)$ the ***dilate*** of f by a.

Let φ be a compactly supported function in $L^2(\mathbb{R}^d)$. Then the ***principal shift invariant (PSI) space*** $V := V(\varphi)$ generated by φ is defined to be the closure in $L^2(\mathbb{R}^2)$ of the set of all finite linear combinations of the shifts of φ:

$$V := V(\varphi) := \overline{\operatorname{span}}\big\{\varphi(\cdot - k) \;\big|\; k \in \mathbb{Z}^d\big\}.$$

For each $j \geq 0$, the space $V_j := V_j(\varphi)$ is defined to be the dilate of V by 2^j, i.e. a function T is in V_j if and only if $T = S(2^j\cdot)$ for $S \in V$. The space V_j is invariant under the shifts $k2^{-j}$, $k \in \mathbb{Z}^d$. Multiresolution adds one essential new ingredient: We require that the spaces V_j are nested, i.e. $V_j \subseteq V_{j+1}$. This is equivalent to $V_0 \subseteq V_1$ which in turn is equivalent to requiring that $\varphi \in V_1$.

Multiresolution analysis (MRA)

In the following we give a definition of multiresolution analysis as can be found in Louis et al. [1994] which is slightly more general than the usual presentations and corresponds to M-filter banks of the Fast Wavelet Transform (FWT), see later. We need a regular integer matrix $A \in \mathbb{Z}^{d\times d}$, called ***dilation matrix***. A real matrix $A \in \mathbb{R}^{d\times d}$ is an integer matrix $A \in \mathbb{Z}^{d\times d}$ if and only if

$$A\mathbb{Z}^d \subseteq \mathbb{Z}^d.$$

That A is ***regular*** in this case means that the determinant $\det A$ (which is in $\mathbb{Z}$) is not zero, i.e. the map

$$A : \mathbb{Z}^d \longrightarrow \mathbb{Z}^d$$

is injective. The image $A\mathbb{Z} \subseteq \mathbb{Z}$ is then called a ***(regular) grid*** in $\mathbb{Z}$. The usual choice is $A = 2$ in the one-dimensional case $d = 1$ and $A = \operatorname{diag}(2,\ldots,2)$ in the multi-dimensional case, as we have chosen it before.

Definition 4.5: *A **multiresolution analysis (MRA)** of $L^2(\mathbb{R}^d)$ is an increasing sequence of closed subspaces $(V_j)_{j\in\mathbb{Z}}$ of $L^2(\mathbb{R}^d)$ having the following three properties:*

(1) *$\bigcap_{-\infty}^{\infty} = \{0\}$ and $\bigcup_{-\infty}^{\infty} V_j$ is dense in $L^2(\mathbb{R}^d)$.*

(2) *For all functions $f \in L^2(\mathbb{R}^d)$ and all integers $j \in \mathbb{Z}$, $f \in V_0$ is equivalent to $f(A^j\cdot) \in V_j$ with a regular matrix (called **dilation matrix**) $A \in \mathbb{Z}^{d\times d}$.*

(3) *There exists a function (called **scaling function**) $\varphi \in V_0$ such that the sequence $\varphi(\cdot - k)$, $k \in \mathbb{Z}^d$, is a Riesz basis for V_0.*

Strang-Fix condition When is a function φ a scaling function, i.e. when does it provide an MRA? Or equivalently, when do the spaces $V_j = V_j(\varphi)$ provide approximation, i.e.

$$\operatorname{dist}(f, V_j)_{L^2(\mathbb{R})} \to 0, \qquad j \to \infty$$

for all $f \in L^2(\mathbb{R})$?

The approximation properties in an MRA are related to polynomial reproduction which can be described by the Fourier transform $\hat{\varphi}$ of φ (Schoenberg, 1946). Strang & Fix (1973) used the Fourier transform to describe approximation properties: φ satisfies the ***Strang-Fix condition*** of order $r \in \mathbb{N}$ if

$$\hat{\varphi}(0) \neq 0 \qquad \text{and} \qquad D^j \hat{\varphi}(2k\pi) = 0, \quad k \in \mathbb{Z}^d \setminus \{0\},\ |j| < r.$$

If φ satisfies the Strang-Fix condition of order r, then $V(\varphi) = V_0(\varphi)$ locally contains all polynomials of order r (degree $< r$).

MRA and wavelets We have a nested sequence

$$\{0\} \subseteq \cdots \subseteq V_{-2} \subseteq V_{-1} \subseteq V_0 \subseteq V_1 \subseteq V_2 \subseteq \cdots \subseteq L^2(\mathbb{R}^d)$$

of subspaces V_j. Especially, $V_0 \subseteq V_1$, and we may consider the orthogonal complement W_0 of V_0 in V_1. One of the main results of wavelet/multiresolution theory is that the orthogonal complement of V_0 in V_1 is an orthogonal sum of PSI spaces generated by wavelets (see e.g. Louis et al. [1994]):

Theorem 4.5 (Meyer)**:** *Let $(V_j)_{j\in\mathbb{Z}}$ be an MRA with dilation matrix A, and let $M := \det A$. Then there exist $M-1$ wavelets $\psi_1, \psi_2, \ldots, \psi_{M-1} \in V_1$ generating an orthonormal basis of the orthogonal complement of V_0 in V_1.*

In some cases it is not necessary to require orthonormal wavelets, instead one is content if they build Riesz bases.

We have then an orthogonal decomposition of V_0 into $M = |\det A|$ subspaces:

$$V_1 = V_0 \oplus \bigoplus_{i=1}^{M-1} W_0^i$$

where the subspaces W_0^i are given by

$$W_0^i := \overline{\operatorname{span}}\{\psi_i(\cdot - k) \mid k \in \mathbb{Z}^d\}.$$

Similar decompositions can be achieved for each $j \in \mathbb{Z}$:

$$V_{j+1} = V_j \oplus \bigoplus_{i=1}^{M-1} W_j^i$$

with

$$W_j^i := \overline{\operatorname{span}}\{\psi_i(A^j \cdot - k) \mid k \in \mathbb{Z}^d\}.$$

The diagram in figure 4.2 summarizes the decompositions of an MRA.

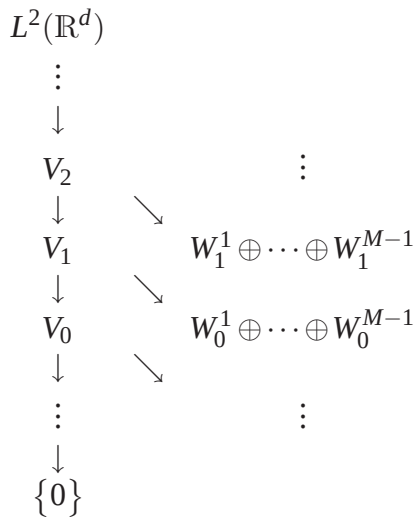


Figure 4.2: Decomposition of $L^2(\mathbb{R}^d)$ in an MRA

Wavelet basis The MRA provides a decomposition

$$L^2(\mathbb{R}^d) = \bigoplus_{j\in\mathbb{Z}} \bigoplus_{i=1}^{M-1} W_j^i.$$

Since we have for each W_j^i the Riesz basis

$$\left\{M^{j/2}\psi_i(A^j x - k) \;\middle|\; k \in \mathbb{Z}^d\right\},$$

we get for $L^2(\mathbb{R}^d)$ the Riesz basis

$$\left\{M^{j/2}\psi_i(A^j x - k) \;\middle|\; i \in \{1,\ldots,M-1\}, j \in \mathbb{Z}, k \in \mathbb{Z}^d\right\},$$

called the corresponding ***wavelet basis***. It is orthonormal if the wavelets are orthonormal in the sense that they provide orthonormal bases of the spaces W_0^i as in Meyer's theorem. If we define the set

$$D := \{1,\ldots,M-1\} \times \mathbb{Z} \times \mathbb{Z}^d,$$

we can write

$$\psi_I(x) = \psi_{i,j,k}(x) = M^{j/2}\psi_i(A^j x - k),$$

for $I = (i,j,k) \in D$. Given a function $f \in L^2(\mathbb{R}^d)$, we thus get a decomposition

$$f = \sum_{I\in D} c_I \psi_I$$

where the scalars $c_I = c_{i,j,k}$ denote the ***wavelet coefficients***.

Wavelet filter From the decomposition

$$V_1 = V_0 \oplus W_0^1 \oplus \cdots \oplus W_0^{M-1},$$

we find that $\varphi, \psi_1, \ldots, \psi_{M-1} \in V_1$, and since the shifts $\varphi(Ax-k)$, $k \in \mathbb{Z}^d$ form a Riesz basis in V_1, we get the following scaling equation

$$\varphi(x) = \sqrt{M} \sum_{k\in\mathbb{Z}} g_k \varphi(Ax-k)$$

for the scaling function φ with the ***low pass filter coefficients*** g_k, and the scaling equations

$$\psi_i(x) = \sqrt{M} \sum_{k\in\mathbb{Z}} h_{i,k} \varphi(Ax-k)$$

for the wavelets ψ_i, $i = 1, \ldots, M-1$, with the ***high pass filter coefficients*** $h_{i,k}$.

According to the definition of an MRA, the shifts $\varphi(\cdot - k)$, $k \in \mathbb{Z}$ of the scaling function φ form a Riesz basis for V_0. If we further assume that the dual basis with respect to that Riesz basis is given by the shifts of a second scaling function $\tilde{\varphi}$ whose dilated spaces $V_j(\tilde{\varphi})$ also form a multiresolution analysis with the same dilation matrix A, then we find further wavelets $\tilde{\psi}_i$, $i = 1, \ldots, M-1$, and we get

$$\tilde{\varphi}(x) = \sqrt{M} \sum_{k\in\mathbb{Z}} \tilde{g}_k \tilde{\varphi}(Ax-k)$$

as well as

$$\tilde{\psi}_i(x) = \sqrt{M} \sum_{k\in\mathbb{Z}} \tilde{h}_{i,k} \tilde{\varphi}(Ax-k).$$

The dual wavelets $\tilde{\psi}_I$ are biorthogonal to the primal wavelets ψ_I, such that each $f \in L^2(\mathbb{R}^d)$ can be written as

$$f = \sum_{I\in D} c_I \psi_I = \langle f, \tilde{\psi}_I \rangle \psi_I$$

with wavelet coefficients

$$c_I := \langle f, \tilde{\psi}_I \rangle,$$

or, written in another way, if $I = (i, j, k)$,

$$c_{i,j,k} := \langle f, \tilde{\psi}_i(A^j \cdot - k) \rangle.$$

If we additionally denote

$$d_{j,k} := \langle f, \tilde{\varphi}(A^j \cdot - k) \rangle$$

for the coefficients with respect to the dual scaling function $\tilde{\varphi}$, then it follows immediately from the scaling equations, that

$$d_{j,k} = \sum_{l\in\mathbb{Z}} \tilde{g}_{l-Ak} d_{j+1,l}$$

and

$$c_{i,j,k} = \sum_{l\in\mathbb{Z}} \tilde{h}_{i,l-Ak} d_{j+1,l}.$$

for $i = 1, \dots, M-1$. We call these M equations the ***decomposition filter bank*** or ***analysis filter bank***. It transforms the coefficients $d_{j+1,l}$ from the scale $j+1$ to the coefficients $d_{j,k}$ and $c_{i,j,k}$ of the scale j. For the opposite direction from scale j to scale $j+1$, it follows from the biorthogonality of the primal und dual scaling function and wavelets that

$$d_{j+1,l} = \sum_{k\in\mathbb{Z}} g_{l-Ak} d_{j,k} + \sum_{i=1}^{M-1} \sum_{k\in\mathbb{Z}} h_{i,l-Ak} c_{i,j,k},$$

called the ***reconstruction filter bank*** or ***synthesis filter bank***.

In the special case that $A = M = 2$, one can show that for $h_k := h_{1,k}$ and $\tilde{h}_k := \tilde{h}_{1,k}$ the relations

$$h_k = (-1)^k g_{1-k} \qquad \text{and} \qquad \tilde{h}_k = (-1)^k \tilde{g}_{1-k}$$

hold.

Wavelet filter and Fast Wavelet Transform (FWT) If we additionally assume that the scaling functions φ and $\tilde{\varphi}$ are compactly supported, then this implies that low and high pass filter coefficients are nonzero only for finitely many values. This property is important for implementations because it allows an exact computation with a finite number of operations. In this case, the algorithm given through the decomposition and reconstruction filter banks works with complexity $O(N)$ if N is the number of non-zero coefficients $d_{j+1,l}$ or the sum of the numbers of non-zero coefficients $d_{j,k}$ and $c_{i,j,k}$, $i = 1, \dots, M-1$, respectively. This algorithm is Mallat's ***Fast Wavelet Transform (FWT)***, and faster than the Fast Fourier Transform (FFT) which is of order $O(N \log N)$ (see e.g. Jaffard et al. [2001]).

Computation of number of coefficients

For implementation issues, we will later need to know how many and which coefficients have to be stored at some scale j. We consider only the case with $A = M = 2$, and denote the scaling and wavelet coefficients of a given function f at a scale $j \in \mathbb{Z}$ by $d_{j,k}$ and $c_{j,k}$, respectively, with $k \in \mathbb{Z}$. For notational convenience, we assume also that the wavelets are orthonormal, such that decomposition and reconstruction filter are equal. Let further be given the filter coefficients of scaling function and wavelet by $(g_m)_{m\in\mathbb{Z}}$ and $(h_m)_{m\in\mathbb{Z}}$, respectively. If we assume that scaling function and wavelet have compact support, then the filter coefficients are almost all zero, only finitely many are non-zero. Let $m_{\min} < m_{\max} \in \mathbb{Z}$ be indices such that all non-zero coefficients are included in

$$(g_m)_{m\in[m_{\min},m_{\max}]} \qquad \text{and} \qquad (h_m)_{m\in[m_{\min},m_{\max}]}.$$

Decomposition For decomposition, we have the formulas

$$d_{j,k} = \sum_{l\in\mathbb{Z}} g_{l-2k} d_{j+1,l}, \qquad c_{j,k} = \sum_{l\in\mathbb{Z}} h_{l-2k} d_{j+1,l}.$$

Assuming that f has also only finitely many non-zero coefficients $c_{j+1,l}$ and $d_{j+1,l}$ at scale $j+1$, say for $l \in [l_{\min}, l_{\max}]$, then we want to compute $k_{\min}$ and $k_{\max}$ such that $d_{j,k}$ and $c_{j,k}$

with $k \in [k_{\min}, k_{\max}]$ include all non-zero coefficients at scale j. Looking at the decomposition formulas above we recognize that the summands can only be non-zero if both

$$l - 2k \in [m_{\min}, m_{\max}] \qquad \text{and} \qquad l \in [l_{\min}, l_{\max}].$$

If we set $m := m(k,l) := l - 2k$ then we see that necessarily $l \equiv m \bmod 2$. Conversely, we have $k = k(m,l) = \frac{l-m}{2}$ whenever $l \equiv m \bmod 2$. From this we get

$$k_{\min} = \min_{\substack{m \in [m_{\min}, m_{\max}] \\ l \in [l_{\min}, l_{\max}]}} k(m,l) = \begin{cases} \frac{l_{\min} - m_{\max}}{2} & \text{if } l_{\min} \equiv m_{\max} \mod 2, \\ \frac{l_{\min} - m_{\max} + 1}{2} & \text{else,} \end{cases}$$

and

$$k_{\max} = \max_{\substack{m \in [m_{\min}, m_{\max}] \\ l \in [l_{\min}, l_{\max}]}} k(m,l) = \begin{cases} \frac{l_{\max} - m_{\min}}{2} & \text{if } l_{\max} \equiv m_{\min} \mod 2, \\ \frac{l_{\max} - m_{\min} - 1}{2} & \text{else.} \end{cases}$$

Reconstruction For reconstruction, we have the formula

$$c_{j+1,l} = \sum_{k \in \mathbb{Z}} g_{l-2k} d_{j,k} + \sum_{k \in \mathbb{Z}} h_{l-2k} c_{j,k}.$$

Assuming now that f has also only finitely many non-zero coefficients $d_{j,k}$ and $c_{j,k}$ at scale j, say for $k \in [k_{\min}, k_{\max}]$, then we want to compute $l_{\min}$ and $l_{\max}$ such that c_l and d_l with $l \in [l_{\min}, l_{\max}]$ include all non-zero coefficients at level $j+1$. Looking at the reconstruction formula above we recognize that the summands can only be non-zero if both

$$l - 2k \in [m_{\min}, m_{\max}] \qquad \text{and} \qquad k \in [k_{\min}, k_{\max}].$$

Setting again $m := m(k,l) := l - 2k$, we have $l = l(m,k) = 2k + m$. From this we get

$$l_{\min} = \min_{\substack{m \in [m_{\min}, m_{\max}] \\ k \in [k_{\min}, k_{\max}]}} l(m,k) = 2k_{\min} + m_{\min}$$

and

$$l_{\max} = \max_{\substack{m \in [m_{\min}, m_{\max}] \\ k \in [k_{\min}, k_{\max}]}} l(m,k) = 2k_{\max} + m_{\max}.$$

Daubechies wavelets

Perhaps the wavelets mostly used in applications are Daubechies wavelets (Daubechies, 1992). They form a family showing the following properties:

- compact support,
- different regularity properties,
- different number of vanishing moments.

Let $A = M = 2$. The orthogonal Daubechies wavelets are indexed according to the positive natural numbers $N = 1, 2, 3, \ldots$, and are only given through their filter coefficients g_k of their scaling functions φ_N (where the filter coefficients of the corresponding wavelet ψ_N can be easily computed by the formula $h_k = (-1)^k g_{1-k}$). The only exception where an explicit formula for Daubechies wavelets can be given is the case $N = 1$: this is the Haar wavelet. For $N \geq 1$, the Daubechies scaling function φ_N and wavelet ψ_N have the following properties:

- φ_N and ψ_N both have filter lengths $2N$ (and thus compact support),
- φ_N and ψ_N are $2N$-times differentiable, and
- ψ_N has N vanishing moments:

$$\int x^n \psi_N(x) dx = 0, \qquad \text{for } n = 0, \ldots, N-1.$$

The filter coefficients of the Daubechies scaling function φ_N are reproduced in table 4.2 (taken from Louis et al. [1994]).

g_k	$N=1$	$N=2$	$N=3$	$N=4$	$N=5$
0	$1/\sqrt{2}$	$\frac{1-\sqrt{3}}{4\sqrt{2}}$	0.332671	0.230378	0.160102
1	$1/\sqrt{2}$	$\frac{1-\sqrt{3}}{4\sqrt{2}}$	0.806892	0.714847	0.603829
2		$\frac{1+\sqrt{3}}{4\sqrt{2}}$	0.459878	0.630881	0.724309
3		$\frac{1+\sqrt{3}}{4\sqrt{2}}$	−0.135011	−0.027984	0.138428
4			−0.085441	−0.187035	−0.242295
5			0.035226	0.030841	−0.032245
6				0.032883	0.077571
7				−0.010597	−0.006241
8					−0.012581
9					0.003336

Table 4.2: Filter coefficients of the Daubechies scaling function

A biorthogonal counterpart was developed by Cohen et al. [1992].

Construction of wavelets in several dimensions

In several dimensions $d \geq 1$, the easiest construction of wavelets is given by ***separable wavelets*** ψ which are built by products of one-dimensional wavelets $\psi_1, \ldots, \psi_d$:

$$\psi(x_1, \ldots, x_d) := \psi_1(x_1) \cdots \psi_d(x_d).$$

This construction goes usually via tensor product wavelets and a diagonal dilation matrix $A = \operatorname{diag}(2, \ldots, 2)$ (see DeVore [1998]):

- Let φ be a univariate scaling function and ψ its corresponding wavelet. Define

  $$\psi^0 := \varphi, \qquad \psi^1 := \psi.$$

  Let E' be the set of the vertices of the unit cube $[0,1]^d$ and E the set of the nonzero vertices. For each vertex $e = (e_1, \ldots, e_d) \in E'$ define the multivariate function

  $$\psi^e(x_1, \ldots, x_d) := \psi^{e_1}(x_1) \cdots \psi^{e_d}(x_d)$$

  and define $\Psi := \left\{ \psi^e \;\middle|\; e \in E \right\}$. If D is the set of indices

  $$D := \left\{ (j,k) \;\middle|\; j \in \mathbb{Z}, k \in \mathbb{Z} \right\},$$

  then

  $$\left\{ \psi_I^e \;\middle|\; I \in D, e \in E \right\}$$

  forms a Riesz basis for $L^2(\mathbb{R}^d)$, and an orthonormal basis if ψ is an orthogonal wavelet. Construct the dual basis functions $\tilde{\psi}_I^e$ with $\tilde{\varphi}$ and $\tilde{\psi}$. Then, each $f \in L^2(\mathbb{R}^d)$ has the wavelet expansion

  $$f = \sum_{I \in D} \sum_{e \in E} c_I^e(f) \psi_I^e, \qquad c_I^e(f) := \langle f, \tilde{\psi}_I^e \rangle.$$

- Another construction is the following: Take the tensor products of the univariate basis ψ_I. This gives the basis

  $$\psi_R(x_1, \ldots, x_d) := \psi_{I_1}(x_1) \cdots \psi_{I_d}(x_d), \qquad R := I_1 \times \cdots \times I_d$$

  where the R's are multidimensional parallelepipeds. Thus, the support of ψ_R corresponds to R and is nonisotropic (can be long in one dimension and short in some other). This is in contrast to the previous construction.

Wavelets in two dimensions In dimension $d = 2$, the tensor product wavelets are given by

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

with $\det A = 4$. We are now looking for choices of A with the minimal value $M := |\det A| = 2$. We follow Louis et al. [1994]. Possible grids $\Gamma := A\mathbb{Z}^d \subseteq \mathbb{Z}^d$ are:

- Row grid $\Gamma = \{(z_1,z_2)^\top \in \mathbb{Z}^2 \mid z_2 \text{ even}\}$,
- Column grid $\Gamma = \{(z_1,z_2)^\top \in \mathbb{Z}^2 \mid z_1 \text{ even}\}$,
- Quincunx grid $\Gamma = \{(z_1,z_2)^\top \in \mathbb{Z}^2 \mid z_1+z_2 \text{ even}\}$.

All grids can be transformed bijectively into another: the row grid into the column grid by mirroring on the diagonal; and if A_1 is the dilation matrix generating the column grid, then

$$A_2 := PA_1P^{-1}$$

with

$$P := \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

generates the Qunincunx grid. Thus, it is enough to examine the Qunincunx grid. Points of the Qunicunx grid are isotropely distributed over $L^2(\mathbb{R}^2)$. The simplest non-equivalent matrices for this grid are:

$$R := \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \qquad \text{and} \qquad S := \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Recall that $D_1, D_2 \in \mathbb{Z}^{d\times d}$ are equivalent if there exists $P \in \mathbb{Z}^{d\times d}$ with $\det P = 1$ (then $P^{-1} \in \mathbb{Z}^{d\times d}$ by Cramer's rule) such that $D_2 = PD_1P^{-1}$.

Construction of biorthogonal wavelets by lifting To be able to construct wavelet filters for arbitrary grids, one has also to choose suitable neighbourhoods in the grid. Nevertheless, the construction of compactly supported biorthogonal wavelets with a prescribed number of vanishing moments in several dimensions resulted to be difficult when tried on the methods used for the construction of one-dimensional wavelets. These methods result in algebraic conditions which are getting very cumbersome in more than three dimensions. Kovačević and Sweldens [2000] proposed a construction avoiding these difficulties. The construction is based on the so-called lifting scheme (introduced by Sweldens [1996] and Sweldens [1997]) which also results in a considerable speed-up compared to the usual filter algorithms. With this construction, multidimensional biorthogonal wavelets for arbitrary dilation matrices A and suitable neighbourhoods can be constructed and the corresponding FWT results in M-filter banks with $M := |\det A|$.

The reason why one is looking for alternatives to tensor product wavelets is that the construction via tensor products of one-dimensional wavelets gives a preference to the coordinate axes and only allows for rectangular divisions in the frequency spectrum (Kovačević and Sweldens [2000]). Often symmetry axes and certain nonrectangular divisions of the frequency spectrum fit better to the applications. Other approaches are either cascade structures or one-to-multidimensional transformations, which either cannot guarantee vanishing moments or perfect reconstruction; in approximation theory, box-splines were used as scaling functions, focussing mostly on low dimensions or separable grids. For further details, see Kovačević and Sweldens [2000].

Wavelets on other spaces

Wavelets on an interval In many practical situations, the functions involved are defined only on a compact set, e.g. the interval $[0,1]$. The application of wavelets in these cases requires modifications: Cohen et al. [1993] obtained necessary boundary corrections to retain orthonormality. Their wavelets also constitute unconditional bases for Besov and Triebel spaces on the interval.

Nevertheless: Usually one works instead with periodic functions g on $\mathbb{R}$ (with unit period). For ***periodic wavelets***, the resolution and spatial indices are restricted to

$$j \geq 0 \qquad \text{and} \qquad k = 0, \ldots, 2^j - 1, \text{respectively.}$$

The DWT respectively the FWT then uses the coefficients periodically, i.e.

$$c_{i,j,k_1} = c_{i,j,k_2} \qquad \text{if } k_1 \equiv k_2 \mod 2^j.$$

Wavelets on other function spaces Wavelets can serve as unconditional bases for a wide range of function spaces, e.g. the L^p spaces, Besov or Triebel spaces. We will return to this issue in section 4.2 after the presentation of the intimate connection between approximation spaces, interpolation spaces, and the connection to the properties of the wavelet coefficients given by corresponding coefficient spaces.

4.1.6 Wavelet packets

Consider the space $\mathscr{L}^2(\mathbb{R}^d)$ and take a scaling function φ and a family of wavelets ψ_i, $i = 1, \ldots, M-1$, such that a multiresolution analysis (V_j) of $\mathscr{L}^2(\mathbb{R}^d)$ is provided with the dilation matrix $A \in \mathbb{Z}^{d\times d}$, and $M := |\det A|$. We saw that we could decompose each V_j into orthogonal spaces V_{j-1} and W_{j-1}^i, $i = 1, \ldots, M-1$. If we decompose not only the spaces V_{j-1} but also the spaces W_{j-1}^i, we get new bases, called wavelet packet bases. This can be done with the same filters as used for the spaces V_j. We will formalize this as follows:

Let $T = (G, r)$, $G = (V, E)$, be an M-ary rooted tree with a strict enumeration q (see chapter 1). To have simpler notations, we will write $\psi_0 := \varphi$. We denote the scaling and wavelet filters with $(h_{i,k})_k$, $i = 0, \ldots, M-1$, for the primal and with $(\tilde{h}_{i,k})_k$, $i = 0, \ldots, M-1$, for the dual filters.

For a given function $\eta \in L^2(\mathbb{R}^d)$, we want to construct new functions using the tree T and the wavelets ψ_i, $i = 0, \ldots, M-1$.

For decomposition, define for each $\eta \in L^2(\mathbb{R}^d)$

$$\tilde{H}_i \eta := \sum_{k\in\mathbb{Z}^d} \tilde{h}_{i,k} \eta(A\cdot - k),$$

und for reconstruction, define for each $\eta_i \in L^2(\mathbb{R}^d)$, $i = 0, \ldots, M-1$,

$$H\eta := \sum_{i=0}^{M-1} \sum_{k\in\mathbb{Z}^d} h_{i,k} \eta_i(A\cdot - k).$$

We have then $\eta = H[\tilde{H}_0\eta,\dots,\tilde{H}_{M-1}]\eta$ and $\eta_i = \tilde{H}_i H[\eta_0,\dots,\eta_{M-1}]$, $i = 0,\dots,M-1$, from biorthogonality. Thus, for a given function η, we can associate to each vertex $v \in V$ recursively a function η_v in the following way:

- Associate $\eta_r := \eta$ to the root r.
- Let v be a vertex, let u be one of its children, and let $i := q_v(u)$ the corresponding enumeration; given η_v, we define η_u by

$$\eta_u = \tilde{H}_i\eta_v.$$

If $b = b(v)$ denotes the associated string to a vertex v of T, we have a one-to-one correspondence of the set V of the vertices and the image $\mathscr{B} = b(V)$. We thus could and will identify V and $\mathscr{B}$, which allows us to wirte η_b instead of η_v if $b = b(v)$. If we denote the concatenation of two strings $b = (b_1,\dots,b_n)$ and $b' = (b'_1,\dots,b'_m)$ by

$$bb' = (b_1,\dots,b_n,b'_1,\dots,b'_m),$$

the rules of decomposition and reconstruction are easily given by

$$\tilde{H}_i\eta_b = \eta_{bi} \qquad \text{and} \qquad \eta_b = H[\eta_{b0},\dots,\eta_{b(M-1)}].$$

Now fix some j_0 and set $\Gamma := V_{j_0}$. Recursively define the spaces

$$\Gamma_{bi} := \tilde{H}_i\Gamma_b$$

for each string $b \in \mathscr{B}$. If we take $\mathscr{B} = \{00\cdots00\} \cup \{1,01,001,0001,\dots,00\cdots01\}$, we are in the situation of the usual MRA decomposition.

If we consider the Riesz basis

$$\gamma := \left\{M^{j_0/2}\varphi(A^{j_0}\cdot - k)\right\}$$

of V_{j_0}, we get recursively bases

$$\gamma_{bi} := \left\{\tilde{H}_i\eta \;\middle|\; \eta \in \gamma_b\right\}$$

as bases of the subspaces

$$\Gamma_{bi} := \overline{\operatorname{span}}\gamma_{bi}$$

such that we have the decompositions of spaces:

$$\Gamma_b = \bigoplus_{i=0}^{M-1} \Gamma_{bi}.$$

At the leaves of the tree T, we have a collection of bases which together form a Riesz basis for V_{j_0}. These bases are called ***wavelet packet bases*** for V_{j_0}. Letting go j_0 to $+\infty$, we get the wavelet packet bases for $L^2(\mathbb{R}^d)$. These are orthonormal if the wavelets chosen are orthonormal.

Haar and Walsh bases as example For the Haar scaling function $\varphi = \mathbf{1}_{[0,1)}$ and the Haar wavelet $\psi = H$, the scaling operators are:

$$\tilde{H}_0\eta := \eta(2\cdot) + \eta(2\cdot - 1), \qquad \tilde{H}_1\eta := \eta(2\cdot) - \eta(2\cdot - 1).$$

If we use the foregoing construction with a binary tree, we get in this particular case of the Haar functions the following special bases:

- The choice $\mathscr{B} = \{00\cdots00\} \cup \{1, 01, 001, 0001, \ldots, 00\cdots01\}$ leads to the Haar basis.
- If we take a tree of some height which is maximally expanded, we get the Walsh basis.

Figure 4.3 shows the corresponding trees.

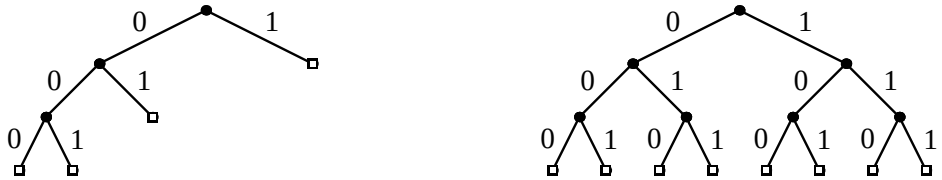


Figure 4.3: Wavelet packet tree leading to Haar basis (left) and Walsh basis (right) (if the heights go to $+\infty$)

Localization of wavelets and wavelet packets We have defined the localization point ω_0 of a function $g \in L^2(\mathbb{R})$ in the frequency domain as the mean value of $|\hat{g}|^2$ (see subsection 4.1.1):

$$\omega_0 := \int_{\mathbb{R}} \omega |\hat{g}(\omega)|^2 d\omega.$$

For a wavelet ψ, this definition should be modified, because for the most usual wavelets the Fourier transformed $\hat{\psi}$ is an even function with a dominant maximum both for positive and for negative frequencies (compare to Louis et al. [1994]). Let therefore be ψ a wavelet with $\|\psi\|_{L^2} = 1$, and let as before

$$t_0 := \int_{\mathbb{R}} t |\psi(t)|^2 dt.$$

Furthermore, define

$$\omega_0^+ := \int_0^\infty \omega |\hat{g}(\omega)|^2 d\omega \qquad \text{and} \qquad \omega_0^- := \int_{-\infty}^0 \omega |\hat{g}(\omega)|^2 d\omega.$$

We then say that ψ localizes at $(t_0, \omega_0^\pm)$. Without loss of generality (possibly after translation), we may assume that $t_0 = 0$. Then

$$\psi_{a,b}(t) := \frac{1}{\sqrt{a}} \psi\Big(\frac{t-b}{a}\Big)$$

localizes at

$$t_0^{ab} = \frac{1}{a} \int_{\mathbb{R}} t \Big|\psi\Big(\frac{t-b}{a}\Big)\Big|^2 dt = b$$

and

$$\omega_0^{\pm ab} = a \int_{0 \le \pm\omega < \infty} \omega |\hat{\psi}(a\omega)|^2 d\omega = \frac{\omega_0^{\pm}}{a}.$$

Thus, with $(a,b) \in \mathbb{R}$, $a \neq 0$, the points $(t_0^{ab}, \omega_0^{\pm ab})$ cover the whole phase space, and one could use as phase space representation

$$(\mathscr{W}_\psi f)(a,b) =: (\mathscr{D} f)\Big(b, \frac{\omega_0^{\pm}}{a}\Big).$$

Thus, for fixed a, the function $(\mathscr{W} f)(a,\cdot)$ represents the variations of the frequencies at $\omega_0^{\pm}/a$ over time, whereas for fixed b the function $(\mathscr{W} f)(\cdot,b)$ represents the frequency distribution at time b. But nevertheless, the localization even for wavelet packets is quite poor. We have for example the following formula (see e.g. [Jaffard et al., 2001]):

$$\limsup_{|b|\to\infty} \left\{ \inf_{\omega\in\mathbb{R}} \int_{-\infty}^{\infty} (\omega - \omega_0^{\pm})^2 |\hat{\gamma}_b|^2 d\omega \right\} = +\infty,$$

i.e. the uncertainty can be arbitrarily large (with the exception of certain γ_b, hence the lim sup).

4.2 Nonlinear approximation

For this section, our main source is [DeVore, 1998].

The fundamental problem of approximation theory is: Resolve a possibly complicated function, called ***target function***, by simpler, easier to compute functions called the approximants. Increasing the resolution of the target function can generally only be achieved by increasing the complexity of the approximants. The understanding of this trade-off between resolution and complexity is the main goal of constructive approximation.

Thus, the goals of approximation theory and of numerical computation are similar. The difference between these two topics lies in the information assumed to be known:

- In approximation theory, one usually assumes that the values of certain simple linear functionals applied to the target function are known.
- In numerical computation, information comes in a less explicit form, e.g. as a solution of an integral equation.

It is impossible to understand numerical computation without understanding of constructive approximation. Developments of constructive approximation and numerical computation followed roughly the same line: Early methods used approximation from finite-dimensional linear spaces, typically spaces of polynomials, both algebraic and trigonometric, until in the late 1950s, there came the development of piecewise polynomials and splines (and their use in Finite Element Methods, FEM).

Shortly thereafter, it was noted that some advantage could be gained by not limiting the approximation to come from linear spaces. In the pioneering work of Birman and Solomyak [1967] on adaptive approximation, no fixed partition for polynomials was used, rather the

partition was allowed to depend on the target function; however, the number of pieces in the approximant is controlled. The idea is simple: Use a finer mesh where the target function is not very smooth, and use a coarser mesh where it is smooth. The problem was how to measure this smoothness. First, exotic spaces were created, then Petrushev [1988] showed that the efficiency of nonlinear spline approximation could be characterized (at least in one variable) by classical smoothness, i.e. Besov spaces. Thus, the advantage of nonlinear approximation became clear.

During the 1980s, multilevel techniques were developed, consisting in parallel developments of multigrid theory (for integral and differential equations), wavelets (harmonic analysis and approximation theory) and multiscale filterbanks (image processing). Wavelets were important on several counts:

- They gave simple and elegant unconditional bases for function spaces (Lebesgue, Hardy, Sobolev, Besov, Triebel-Lizorkin) that simplified some aspects of Littlewood-Paley theory.
- They provided a good vehicle of core linear operators of harmonic analysis and partial differential equations (Calderón-Zygmund theory).
- They allowed the solution of functional analytic and statistical extremal problems to be made directly from wavelet coefficients.

Wavelet theory provides simple and powerful decompositions of the target function into a series of building blocks. Thus, one can approximate the target function by selecting certain terms of this series.

- Taking partial sums of this series yields approximation from linear spaces: It was easy to establish that this form of linear approximation offered little advantage over spline methods.
- Letting the selection of terms to be chosen from the wavelet series depend on the target function and keeping control only over the number of the terms to be used: This form of nonlinear approximation is called ***n*-term approximation**.

n-term approximation was introduced by Schmidt [1907] and much later utilized for multivariate splines by Oskolkov [1979].

Most function norms can be described in terms of wavelet coefficients. This simplifies the characterization of functions with a specified approximation order and makes transparent strategies for achieving good or best *n*-term approximations: It is enough to retain the *n* terms in the wavelet expansion of the target function that are largest relative to the norm measuring the error of approximation. Thus, it is enough to threshold the properly normalized wavelet coefficients. This leads to approximation strategies based on wavelet shrinkage (Donoho and Johnstone [1994]). This in turn was used to solve several extremal problems in statistical estimation, e.g. the recovery of the target function in the presence of noise.

Wavelets are tailor-made for nonlinear approximation and certain numerical computations:

- Computation is fast and simple.

- Strategies for generating good nonlinear approximation are transparent.
- Wavelets provide unconditional bases for many function spaces and smoothness spaces.
- Thus characterization of approximation is greatly simplified.
- Wavelets generalize readily to several dimensions.

The next step in nonlinear approximation is to try to incorporate the choice of the basis into the approximation problem. We have a double-stage nonlinear approximation problem:

- Use the target function to choose a good or best basis from a given library of bases, and
- then choose the best n-term approximation relative to this good basis.

This is a form of ***highly nonlinear approximation***. Other examples provide the greedy algorithms and adaptive pursuit for finding an n-term approximation from a redundant set of functions (called dictionary). The understanding of highly nonlinear methods is quite fragmentary. Describing functions that have a specified rate of approximation remains a challenging problem.

General notations Constants appearing in equations are simply denoted by C and may vary at each occurrence, even in the same formula. Sometimes the parameters on which constants depend will be indicated: $C(p)$ or $C(p,\alpha)$.

Another notation often used is the following:

$$A \asymp B$$

which means there are constants $C_1, C_2 > 0$ such that

$$C_1 A \leq B \leq C_2 A$$

where A and B are two expressions depending on other variables (parameters). If the expressions A and B denote (semi-)norms, the notation simply means the equivalence of these (semi-) norms.

4.2.1 Approximation theory

The interplay of three types of spaces shows to be extremely fruitful:

- approximation spaces,
- interpolation spaces, and
- smoothness spaces.

These three topics are intimately connected, giving insight into how to solve the approximation problem (DeVore [1998]).

Basic definitions

We follow DeVore [1998]. Be given a normed space $(X, \|\cdot\|_X)$ in which approximation takes place and a ***target function*** $f \in X$ which is to be approximated. Further be given spaces $X_n \subseteq X$, $n = 0, 1, \ldots$ and ***approximants*** $g \in X_n$. Then the ***approximation error*** is defined as:

$$E_n(f)_X := \operatorname{dist}(f, X_n)_X := \inf_{g \in X_n} \|f - g\|_X.$$

In ***linear approximation***, the X_n are vector spaces, and usually n is the dimension of X_n. In ***nonlinear approximation***, the X_n can be quite general and do not have to be linear; n relates to the number of free parameters.

We make the following assumptions:

(i) $X_0 := \{0\}$,

(ii) $X_n \subseteq X_{n+1}$,

(iii) $aX_n = X_n$, $a \in \mathbb{R}$, $a \neq 0$,

(iv) $X_n + X_n \subseteq X_{cn}$ for some integer constant $c \geq 1$ independent of n,

(v) each $f \in X$ has a best approximation from X_n,

(vi) $\lim_{n\to\infty} E_n(f)_X = 0$ for all $f \in X$.

The most essential ones are (iii), (iv), (vi), the others are more made for convenience and can be eliminated or modified with a similar theory. From (ii) and (vi) it follows that $E_n(f)_X$ decreases monotonically to 0 as $n \to \infty$.

Running example: Hilbert spaces The concepts of approximation theory are most easily seen in Hilbert spaces (see DeVore [1998]):

Example: Let $\mathscr{H}$ be a separable Hilbert space with inner product $\langle \cdot, \cdot \rangle$ and norm $\|\cdot\|_{\mathscr{H}}$ and let η_k, $k = 1, 2, \ldots$, be an orthonormal basis.

- Linear approximation: Use linear spaces

  $$\mathscr{H}_n := \operatorname{span}\{\eta_k \mid 1 \leq k \leq n\}$$

  to approximate an element $f \in \mathscr{H}$. The approximation error is measured by

  $$E_n(f)_{\mathscr{H}} := \inf_{g \in \mathscr{H}_n} \|f - g\|_{\mathscr{H}}.$$

- n-term approximation: Replace the spaces $\mathscr{H}_n$ by Σ_n consisting of all elements $g \in \mathscr{H}$ that can be expressed by

  $$g \in \sum_{k \in \Lambda} c_k \eta_k$$

where $\Lambda \subseteq \mathbb{N}$ is a set of indices with $\#\Lambda \leq n$. The spaces Σ_n are not linear: A sum of two elements of Σ_n will in general need $2n$ terms in its representation by the η_k. In this example, we reserve the notation E_n for the linear approximation error and denote the error in the case of n-term approximation by σ_n. We thus have the ***error of n-term approximation***

$$\sigma_n(f)_{\mathscr{H}} := \inf_{g \in \Sigma_n} \|f - g\|_{\mathscr{H}}.$$

Approximation spaces

We still use DeVore [1998]. The question that arises is: Which functions $f \in X$ can be approximated at a given rate like $O(n^{-\alpha})$? We will let $\mathscr{A}^\alpha := \mathscr{A}^\alpha(X,(X_n))$ consist of all functions $f \in X$ for which

$$E_n(f)_X = O(n^{-\alpha}) \qquad n \to \infty.$$

or, put more concretely,

$$E_n(f)_X \leq Mn^{-\alpha}, \qquad n = 1,2,\ldots$$

for some constant $M > 0$. We define then

$$|f|_{\mathscr{A}^\alpha(X,(X_n))}$$

as the infimum of all such M. The goal is to characterize $\mathscr{A}^\alpha$.

Sometimes finer statements about the decrease of the error $E_n(f)_X$ are needed: For each $\alpha > 0$ and $0 < q < \infty$ we define the ***approximation space***

$$\mathscr{A}_q^\alpha := \mathscr{A}_q^\alpha(X,(X_n))$$

as the set of all functions $f \in X$ such that

$$|f|_{\mathscr{A}_q^\alpha} := \begin{cases} \left(\sum_{n=1}^\infty \left[n^\alpha E_n(f)_X\right]^q \frac{1}{n}\right)^{1/q}, & 0 < q < \infty, \\ \sup_{n \geq 1} n^\alpha E_n(f)_X, & q = \infty, \end{cases}$$

is finite. Further define $\|f\|_{\mathscr{A}_q^\alpha} := |f|_{\mathscr{A}_q^\alpha} + \|f\|_X$.

Thus, the case $q = \infty$ is the space $\mathscr{A}^\alpha$. For $q < \infty$, the requirement for membership in $\mathscr{A}_q^\alpha$ gets stronger as q decreases:

$$\mathscr{A}_q^\alpha \subseteq \mathscr{A}_p^\alpha, \qquad 0 < q < p \leq \infty.$$

However, all of these spaces correspond to a decrease in error like $O(n^{-\alpha})$. Because of the monotonicity of the sequence $(E_n(f)_X)$, we have the equivalence

$$|f|_{\mathscr{A}_q^\alpha} \asymp \begin{cases} \left(\sum_{k=0}^\infty \left[2^{k\alpha} E_{2^k}(f)_X\right]^q\right)^{1/q}, & 0 < q < \infty, \\ \sup_{k \geq 0} 2^{k\alpha} E_{2^k}(f)_X, & q = \infty, \end{cases}$$

which is usually more convenient to work with.

Linear and nonlinear approximation in Hilbert space We come back to the example of Hilbert spaces (DeVore [1998]):

Example (Hilbert spaces continued)**:** We want to describe the approximation classes in terms of coefficients of the orthogonal expansion

$$f = \sum_{k=1}^{\infty} \langle f, \eta_k \rangle \eta_k = \sum_{k=1}^{\infty} f_k \eta_k$$

where we set

$$f_k := \langle f, \eta_k \rangle, \qquad k = 1, 2, \ldots .$$

- Case of linear approximation: The best approximation to f from $\mathscr{H}_n$ is given by the projection

  $$P_n f := \sum_{k=1}^{n} f_k \eta_k$$

  onto $\mathscr{H}_n$ and the approximation error satisfies

  $$E_n(f)^2_{\mathscr{H}} = \sum_{k=n+1}^{\infty} |f_k|^2.$$

  We can characterize $\mathscr{A}^\alpha$ in terms of the dyadic sums

  $$F_m := \left(\sum_{k=2^{m-1}+1}^{2^m} |f_k|^2 \right)^{1/2}, \qquad m = 1, 2, \ldots .$$

  Indeed: $f \in \mathscr{A}^\alpha((\mathscr{H}_n))$ if and only if

  $$F_m \le M 2^{-m\alpha}, \qquad m = 1, 2, \ldots$$

  and the smallest M is equivalent to $\|f\|_{\mathscr{A}^\alpha((\mathscr{H}_n))}$.

  Let us consider a variant of $\mathscr{A}^\alpha$: Let $\mathscr{A}_2^\alpha((\mathscr{H}_n))$ denote the set of all f such that

  $$|f|_{\mathscr{A}_2^\alpha((\mathscr{H}_n))} := \left(\sum_{n=1}^{\infty} [n^\alpha E_n(f)_{\mathscr{H}}]^2 \frac{1}{n} \right)^{1/2}$$

  is finite. From the monotonicity of $E_k(f)_{\mathscr{H}}$, it follows

  $$|f|_{\mathscr{A}_2^\alpha((\mathscr{H}_n))} \asymp \left(\sum_{k=0}^{\infty} 2^{2k\alpha} E_{2^k}(f)^2_{\mathscr{H}} \right)^{1/2}.$$

  The condition for membership in $\mathscr{A}_2^\alpha$ is slightly stronger than membership in $\mathscr{A}^\alpha$: the latter requires that the sequence $(n^\alpha E_n)$ is bounded while the former requires that it is square summable with weight $1/n$. The space $\mathscr{A}_2^\alpha((\mathscr{H}_n))$ is characterized by

  $$\sum_{k=1}^{\infty} k^{2\alpha} |f_k|^2 \le M^2$$

  and the smallest such M is equivalent to $|f|_{\mathscr{A}_2^\alpha((\mathscr{H}_n))}$.

- Case of nonlinear approximation: We can characterize the space $\mathscr{A}^\alpha((\Sigma_n))$ by using the rearrangement of the coefficients f_k. Denote by $\gamma_k(f)$ the k-th largest of the numbers $|f_j|$. Then: $f \in \mathscr{A}^\alpha((\Sigma_n))$ if and only if

$$\gamma_n(f) \leq Mn^{-\alpha-1/2}$$

and the infimum of all M is equivalent to $|f|_{\mathscr{A}^\alpha((\Sigma_n))}$.

Interpolation spaces

We follow again DeVore [1998]. Given two spaces X and Y continuously contained in some larger space, for which spaces Z is it true that each linear operator T mapping X and Y boundedly into themselves automatically maps Z boundedly into itself? Such spaces are called ***interpolation spaces*** for the pair (X,Y). The task is to construct and to characterize such spaces for a given pair (X,Y).

Example (Interpolation spaces for (L^1,L^∞))**:**

- The Riesz-Thorin theorem states that L^p, $1 < p < \infty$ are interpolation spaces.
- The Calderón-Mitjagin theorem characterizes all interpolation spaces for this pair as the rearrangement-invariant function spaces.

There exist two primary methods for the construction of interpolation spaces: The complex method developed by Calderón [1964] and the real method of Lions and Peetre (Peetre [1963]).

Real interpolation We describe only the real method of Lions and Peetre. Let (X,Y) be a pair of normed linear spaces, and assume that Y is continuously embedded in X, i.e.

$$Y \subseteq X \qquad \text{and} \qquad \|\cdot\|_X \leq C\|\cdot\|_Y.$$

For any $t > 0$, define the ***K-functional***

$$K(f,t) := K(f,t;X,Y) := \inf_{g\in Y} \|f-g\|_X + t|g|_Y$$

where $\|\cdot\|_X$ is the norm in X and $|\cdot|_Y$ is a semi-norm on Y or even a quasi-semi-norm, where the triangle inequality is replaced by

$$|g_1+g_2|_Y \leq C(|g_1|_Y + |g_2|_Y)$$

with an absolute constant C. The K-functional $K(f,\cdot)$ is defined on $\mathbb{R}_{\geq 0}$, and monotone and concave.

Let T be a linear operator which maps X and Y into themselves with a norm bounded by M in both cases. Then: For any $g \in Y$, we have

$$Tf = T(f-g) + Tg$$

and therefore

$$K(Tf,t) \le \|T(f-g)\|_X + t|Tg|_Y \le M(\|f-g\|_X + t|g|_Y).$$

Taking the infimum over g, we get

$$K(Tf,t) \le MK(f,t), \qquad t>0.$$

For any function norm $\|\cdot\|$ defined for real valued functions on $\mathbb{R}_{\ge 0}$ we obtain

$$\|K(Tf,\cdot)\| \le M\|K(f,\cdot)\|.$$

The space of functions f for which $\|K(f,\cdot)\|$ is finite will be an interpolation space.

θ,q-norms The most common choice of the norm $\|\cdot\|$ used for interpolation are the θ,q-norms. They are defined analogous to the norms used for defining approximation spaces: For $0<\theta<1$ and $0<q\le\infty$, the interpolation space $(X,Y)_{\theta,q}$ is defined as the set of all functions $f\in X$ such that the ***θ,q-norm***

$$|f|_{(X,Y)_{\theta,q}} := \begin{cases} \left(\int_0^\infty \left[t^{-\theta}K(f,t)\right]^q \frac{dt}{t}\right)^{1/q}, & 0<q<\infty, \\ \sup_{t>0} t^{-\theta}K(f,t), & q=\infty, \end{cases}$$

is finite. The repeated application of the construction with θ,q-norms brings nothing new (see e.g. DeVore [1998]):

Theorem 4.6 (*Reiteration theorem*): *Let $X' := (X,Y)_{\theta_1,q_1}$ and $Y' := (X,Y)_{\theta_2,q_2}$. Then, for all $0<\theta<1$ and $0<q\le\infty$, we have*

$$(X',Y')_{\theta,q} = (X,Y)_{\alpha,q}, \qquad \alpha := (1-\theta)\theta_1 + \theta\theta_2.$$

There is the following simplification of the θ,q-norm:

1) Using the fact that Y is continuously embedded in X, we obtain an equivalent norm by taking the integral over $[0,1]$ in the definition of the θ,q-norm.

2) Since $K(f,\cdot)$ is monotone, the integral can be discretized.

This gives:

$$|f|_{(X,Y)_{\theta,q}} \asymp \begin{cases} \left(\sum_{k=0}^\infty \left[2^{k\theta}K(f,2^{-k})\right]^q\right)^{1/q}, & 0<q<\infty, \\ \sup_{k\ge 0} 2^{k\theta}K(f,2^{-k}), & q=\infty. \end{cases}$$

In this form, the definitions of approximation and interpolation spaces are almost identical: We have replaced E_{2^k} by $K(f,2^{-k})$. Our aim is to characterize the one by the other, but we need a comparison between the error $E_n(f)$ and the K-functional K. This can only be achieved if we make the right choice of Y.

Jackson and Bernstein inequalities We still follow DeVore [1998]. To be able to compare approximation and interpolation spaces, two inequalities play a major role: the Jackson and the Bernstein inequality. Let $r > 0$. Then the ***Jackson inequality*** or ***direct theorem*** is given by:

$$E_n(f)_X \leq Cn^{-r}|f|_Y, \qquad f \in Y, \qquad n = 1,2,\ldots,$$

whereas the corresponding ***Bernstein inequality*** or ***inverse theorem*** is:

$$|S|_Y \leq Cn^r\|S\|_X, \qquad S \in X_n, \qquad n = 1,2,\ldots.$$

In both cases, $C = C(r)$ is only depending on r. We have the following theorem (see e.g. DeVore [1998]):

Theorem 4.7: *If the Jackson and Bernstein inequalities are valid, then for each $0 < \gamma < r$ and $0 < q \leq \infty$ the following relation holds between approximation spaces and interpolation spaces:*

$$\mathscr{A}_q^\gamma(X) = (X,Y)_{\gamma/r,q}$$

with equivalent norms.

Therefore, this theorem solves the task of characterizing the approximation spaces if we know two ingredients:

(i) an appropriate space Y for which the Jackson and Bernstein inequalities hold, and

(ii) a characterization of the interpolation spaces $(X,Y)_{\theta,q}$.

Approximation spaces as interpolation spaces The approximation spaces are actually interpolation spaces (see DeVore [1998]):

Theorem 4.8 (DeVore and Popov [1988])**:** *For any space X and spaces X_n, as well as for any $r > 0$ and $0 \leq s \leq \infty$, the spaces X_n, $n = 1,2,\ldots$, satisfy the Jackson and Bernstein inequalities for $Y = \mathscr{A}_s^r(X)$. Therefore, for any $0 < \alpha < r$ and $0 < q \leq \infty$, we have*

$$\mathscr{A}_q^\alpha(X) = (X, \mathscr{A}_s^r(X))_{\alpha/r,q}.$$

In other words: The approximation family $\mathscr{A}_q^\alpha(X)$ is an interpolation family.

Approximation can in turn also be used to characterize interpolation spaces (see DeVore [1998]):

Definition 4.6: ***(i)*** *A sequence (T_n), $n = 1,2,\ldots$, of (possibly nonlinear) operators T_n mapping X into X_n provides **near best approximation** if there is an absolute constant $C > 0$ such that*

$$\|f - T_n f\|_X \leq CE_n(f)_X, \qquad n = 1,2,\ldots.$$

(ii) *This family is **stable on** Y if there is an absolute constant $C > 0$ such that*

$$|T_n f|_Y \leq C|f|_Y, \qquad n = 1,2,\ldots.$$

Theorem 4.9: *Let $X, Y, (X_n)$ be as above and suppose that (X_n) satisfies the Jackson and Bernstein inequalities; suppose further that the sequence of operators (T_n) provides near best approximation and is stable on Y. Then T_n realizes the K-functional, i.e. there is an absolute constant $C > 0$ such that*

$$\|f - T_n f\|_X + n^{-r}|T_n f|_Y \le CK(f, n^{-r}, X, Y).$$

Interpolation for L^1, L^∞: Lorentz spaces We follow again DeVore [1998]. Let $(A, d\mu)$ be a sigma-finite measure space and consider the pair $(L^1(A,d\mu), L^\infty(A,d\mu))$. Consider the ***decreasing rearrangement*** f^* of a μ-measurable function f: f^* is a nonnegative, nonincreasing function defined on $\mathbb{R}_{\ge 0}$ which is equimeasurable with f:

$$\mu(f,t) := \mu\big(\{x \mid |f(x)| > t\}\big) = \Big|\{s \mid f^*(s) > t\}\Big|, \qquad t > 0$$

($|E|$ denoting the Lebesgue measure of a set E). The rearrangement f^* can be defined directly via:

$$f^*(t) := \inf\{y \mid \mu(f,t) \le y\}.$$

Thus f^* is essentially the inverse function to $\mu(f,t)$.

There is the following formula for the K-functional involving the rearrangement f^* of a function f (DeVore and Lorentz [1993]):

$$K(f,t,L^1,L^\infty) = \int_0^t f^*(s)ds \qquad \text{for all } f \in L^1 + L^\infty.$$

From the fact that

$$\int_A |f|^p d\mu = \int_0^\infty (f^*(s))^p ds,$$

it is easy to deduce from this formula the Riesz-Thorin theorem for this pair.

With this K-functional, it is easy to describe the (θ, q) interpolation spaces in terms of Lorentz spaces: For each $0 < p < \infty$, $0 < q \le \infty$, the ***Lorentz space*** $L^{p,q}(A, d\mu)$ is defined by the set of all μ-measurable functions f such that

$$\|f\|_{L^{p,q}} := \begin{cases} \left(\int_0^\infty \left[t^{1/p} f^*(t)\right]^q \frac{dt}{t}\right)^{1/q}, & 0 < q < \infty, \\ \sup_{t>0} t^{1/p} f^*(t), & q = \infty, \end{cases}$$

is finite. Replacing f^* by $\frac{1}{t}\int_0^t f^*(s)ds = K(f,t)/t$ and using the Hardy inequalities one obtains

$$(L^1(A,d\mu), L^\infty(A,d\mu))_{1-1/p,q} = L^{p,q}(A,d\mu), \qquad 1 < p < \infty,\ 0 < q \le \infty.$$

The space $L^{p,\infty}$ is better known as ***weak*** L^p and can be equivalently defined by the condition

$$\mu\big(\{x \mid |f(x)| > y\}\big) \le M^p y^{-p}.$$

The smallest M for which this is valid is equivalent to the norm in $L^{p,\infty}$. The above results include the case where $d\mu$ is a mixture of point masses (delta distributions), for example if

$d\mu$ is the counting measure on $\mathbb{N}$. Especially in this case we use the following notation: Let $\ell^p := \ell^p(\mathbb{N})$ the collection of sequences $x = (x(n))_{n\in\mathbb{N}}$ for which

$$\|x\|_{\ell^p} := \begin{cases} \left(\sum_{n=0}^{\infty} |x(n)|^p\right)^{1/p}, & 0 < p < \infty, \\ \sup_{n\in\mathbb{N}} |x(n)|, & p = \infty, \end{cases}$$

is finite. Then $\ell^p(\mathbb{N}) = L^p(\mathbb{N}, d\mu)$ where μ is the counting measure. We denote the Lorentz spaces in this case by $\ell^{p,q}$. The space $\ell^{p,\infty}$ (weak ℓ^∞) thus consists of all sequences that satisfy

$$x^*(n) \le Mn^{-1/p}$$

with $(x^*(n))$ the decreasing rearrangement of $(|x(n)|)$, equivalently stated as

$$\#\{n \mid |x(n)| > y\} \le M^p y^{-p}.$$

The interpolation theory for L^p applies for more than the pair (L^1, L^∞). We give the formulation only for spaces $\ell^{p,q}$: For any $0 < p_1 < p_2 < \infty$, $0 < q_1, q_2 \le \infty$, we have

$$(\ell^{p_1,q_1}, \ell^{p_2,q_2})_{\theta,q} = \ell^{p,q}, \qquad \frac{1}{p} := \frac{1-\theta}{p_1} + \frac{\theta}{p_2}, \qquad 0 < q \le \infty$$

with equivalent norms. For $1 \le p_1, p_2 \le \infty$ this follows from the reiteration theorem, the general case needs slight modifications (see Bergh and Löfström [1976]).

Smoothness spaces

We use again DeVore [1998]. There are two important ways to describe smoothness spaces:

(i) Through notions like differentiability and moduli of smoothness; most smoothness spaces were originally introduced in this fashion.

(ii) Through expansion of functions into a series of building blocks (e.g. Fourier or wavelet) and describing smoothness as decay condition on the coefficients in such expansions.

That these descriptions are equivalent is at the heart of the subject (DeVore [1998]).

Sobolev spaces Let $1 \le p \le \infty$, let $r > 0$ be an integer, and let $\Omega \subseteq \mathbb{R}^d$ be a domain (here: open, connected set). The ***Sobolev space*** $W^{r,p}(\Omega) := W^r(L^p(\Omega))$ is defined as the set of all measurable functions f defined on Ω which have all their distributional derivatives $D^\nu f$, $|\nu| \le r$, in $L^p(\Omega)$. Here, we write

$$|\nu| := |\nu_1| + \cdots + |\nu_d|$$

for a multiindex $\nu = (\nu_1, \ldots, \nu_d)$, $\nu_i \in \mathbb{N}$. The semi-norm for $W^r(L^p(\Omega))$ is defined by

$$|f|_{W^r(L^p(\Omega))} := \sum_{|\nu|=r} \|D^\nu f\|_{L^p(\Omega)}$$

and their norm by

$$\|f\|_{W^r(L^p(\Omega))} = |f|_{W^r(L^p(\Omega))} + \|f\|_{L^p(\Omega)}.$$

Thus, Sobolev spaces measure smoothness of order r in L^p when r is a positive integer and $1 \le p \le \infty$. A deficiency is that this definition does not immediately apply when r is non-integral or when $p < 1$.

Differences and moduli of smoothness We follow DeVore [1998]. One way to derive smoothness of fractional order is through differences. For $h \in \mathbb{R}^d$, let T_h denote the ***translation operator***, defined by

$$T_h f := f(\cdot + h) \qquad \text{for a function } f,$$

and let denote I the identity operator. Then, for any positive integer r,

$$\Delta_h^r := (T_h - I)^r$$

is the r***-th difference operator with step*** h. Clearly,

$$\Delta_h^r = \Delta_h^1(\Delta_h^{r-1})$$

and also, for a function f on Ω,

$$\Delta_h^r(f)(x) = \sum_{k=0}^{r} (-1)^{r-k} \binom{r}{k} f(x+kh)$$

with the convention that $\Delta_h^r(f)(x)$ is defined to be zero if any of the points

$$x, \ldots, x+rh$$

is not in Ω.

We can use Δ_r^h to measure smoothness. If $f \in L^p(\Omega)$, $0 < p \le \infty$, then

$$\omega_r(f,t)_p := \sup_{|h| \le t} \|\Delta_h^r(f)\|_{L^p(\Omega)}$$

is the r***-th order modulus of smoothness*** of f in $L^p(\Omega)$. In the case $p = \infty$, $L^\infty(\Omega)$ is replaced by $C^0(\Omega)$, the space of uniformly continuous functions on Ω. We always have that

$$\omega_r(f,t)_p \to 0 \text{ monotonically as } t \to 0.$$

The faster this convergence to 0, the smoother is f.

Smoothness spaces are created by bringing together all functions whose moduli of smoothness have a common behaviour.

Besov spaces We have three parameters in the description of Besov spaces (see e.g. DeVore [1998]):

- Two primary parameters: α giving the order of smoothness (for instance the number of derivatives) and p giving the L^p space in which smoothness is measured.
- One secondary parameter q that allows subtle distinctions.

Let $\alpha > 0$, $0 < p \le \infty$, $0 < q \le \infty$, $r := \lfloor \alpha \rfloor + 1$ (i.e., the smallest integer larger than α). We say that f is in the ***Besov space*** $B_{p,q}^\alpha(\Omega) := B_q^\alpha(L^p(\Omega))$ if

$$|f|_{B_q^\alpha(L^p(\Omega))} := \begin{cases} \left(\int_0^\infty \left[t^{-\alpha} \omega_r(f,t)_p \right]^q \frac{dt}{t} \right)^{1/q}, & 0 < q < \infty, \\ \sup_{t>0} t^{-\alpha} \omega_r(f,t)_p, & q = \infty, \end{cases}$$

is finite. This is the semi-norm in $B_q^\alpha(L^p(\Omega))$. The ***Besov norm*** is given by

$$\|f\|_{B_q^\alpha(L^p(\Omega))} := |f|_{B_q^\alpha(L^p(\Omega))} + \|f\|_{L^p(\Omega)}.$$

We have thus a complete analogy to approximation spaces and interpolation spaces.

Besov spaces and their relation to other spaces We want to assume now that the domain Ω is a ***Lipschitz domain***, i.e. the boundary $\partial\Omega$ of Ω is Lipschitz (see appendix).

- The number r was taken to be the smallest integer larger than α; actually any integer $r > \alpha$ will define the same space with equivalent norm.
- If $\alpha < 1$ and $q = \infty$, than $B^\alpha_\infty(L^p(\Omega)) = \mathrm{Lip}(\alpha, L^p(\Omega))$ with identical semi-norm and norm.
- If $\alpha = 1$, then $B^1_\infty(L^p(\Omega)) \supsetneqq \mathrm{Lip}(1, L^p(\Omega))$ because $B^1_\infty(L^p(\Omega))$ uses ω_2 in the definition whereas $\mathrm{Lip}(1, L^p(\Omega))$ uses ω_1 and
$$\omega_2(f,t)_p \le 2^{\max(1/p,1)}\omega_1(f,t)_p.$$
- For the same reason: $B^r_\infty(L^p(\Omega)) \supsetneqq W^r(L^p(\Omega))$ for $1 \le p \le \infty$, $p \ne 2$, and r an integer (the Sobolev space could be described by replacing ω_{r+1} by ω_r in the definition of the Besov space).
- For $p = 2$ and r again an integer, we have $B^r_2(L^2(\Omega)) = W^r(L^2(\Omega))$, the Sobolov space.

The Sobolev embedding theorem Increasing the secondary index q gives a larger space (however distinctions are small):

$$B^\alpha_{q_1}(L^p(\Omega)) \subsetneqq B^\alpha_{q_2}(L^p(\Omega)) \qquad \text{for } q_1 < q_2.$$

The ***Sobolev embedding theorem*** is easiest described pictorially, see figure 4.4 (see e.g. DeVore [1998]). Identify the Besov space with primary indices p and α with the point $(1/p, \alpha)$ in the upper right quadrant of $\mathbb{R}^2$. Then the line with slope d going through $(1/p, 0)$ is the demarcation line for embeddings of Besov spaces into $L^p(\Omega)$: Any Besov space with primary indices corresponding to points

- above that line is embedded in $L^p(\Omega)$, regardless of q,
- on the line may or may not be embedded in $L^p(\Omega)$, e.g. $B^\alpha_\tau(L^\tau(\Omega))$ are, where
$$1/\tau = \alpha/d + 1/p,$$
- below that line is never embedded in $L^p(\Omega)$.

Interpolation of smoothness spaces Interpolation between $L^p(\Omega)$ and a Sobolev space $W^r(L^p(\Omega))$ (see DeVore [1998]): For the K-functional holds

$$K\big(f, t^r, L^p(\Omega), W^r(L^p(\Omega))\big) \asymp \omega_r(f,t)_p.$$

Thus:
$$\big(L^p(\Omega), W^r(L^p(\Omega))\big)_{\theta,q} = B^{\theta r}_q(L^p(\Omega)), \qquad 0 < \theta < 1, \quad 0 < q \le \infty,$$

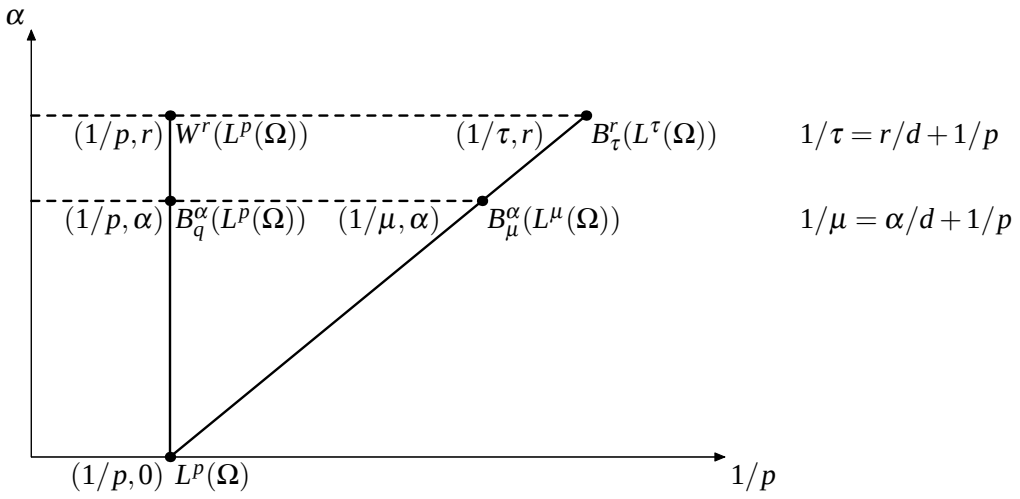


Figure 4.4: Linear and nonlinear approximation

with equivalent norms. From the reiteration theorem, it follows that, for $\alpha_1 < \alpha_2$ and any $0 < q_1, q_2 \le \infty$, we have for any $0 < \theta < 1$, $0 < q \le \infty$:

$$\left(B^{\alpha_1}_{q_1}(L^p(\Omega)), B^{\alpha_2}_{q_2}(L^p(\Omega))\right)_{\theta,q} = B^{\alpha}_{q}(L^p(\Omega)), \qquad \alpha := (1-\theta)\alpha_1 + \theta\alpha_2.$$

We can replace $B^{\alpha_1}_{q_1}(L^p(\Omega))$ by $L^p(\Omega)$ and obtain for any $0 < r \le \infty$:

$$\left(L^p(\Omega), B^{\alpha}_{r}(L^p(\Omega))\right)_{\theta,q} = B^{\theta\alpha}_{q}(L^p(\Omega)), \qquad 0 < \theta < 1, \quad 0 < q \le \infty.$$

Fix a value $p \in (0,\infty)$ and consider Besov spaces $B^{\alpha}_{\tau}(L^\tau(\Omega))$ where τ and α are related by

$$\frac{1}{\tau} = \frac{\alpha}{d} + \frac{1}{p}.$$

(These spaces correspond to points on the line segment with slope d passing through $(1/p, 0)$ corresponding to $L^p(\Omega)$). We have the following interpolation result:

$$\left(L^p(\Omega), B^{\alpha}_{\tau}(L^\tau(\Omega))\right)_{\theta,q} = B^{\theta\alpha}_{q}(L^q(\Omega)), \qquad \text{provided } \frac{1}{q} = \frac{\theta\alpha}{d} + \frac{1}{p}.$$

This means, if we interpolate between two Besov spaces corresponding to points on this line, we get another Besov space corresponding to a point on this line, provided we choose the secondary indices in a suitable way.

Example: Hilbert space We continue our example of Hilbert spaces (following DeVore [1998]):

Example (Hilbert spaces continued)**:** Nonlinear n-term approximation in Hilbert space $\mathscr{H}$: We could characterize $\mathscr{A}^r_\infty((\mathscr{H}_n))$ for any $r > 0$ by the condition

$$\gamma_n(f) \le Mn^{-r-1/2},$$

with $\gamma_n(f)$ the rearranged coefficients. We see now: The sequence $f_k := \langle f, \eta_k \rangle$ is in weak $\ell^{\tau(r)}$ ($= \ell^{\tau(r),\infty}$) with $\tau(r)$ defined by

$$\frac{1}{\tau(r)} = r + \frac{1}{2}.$$

The smallest M for which this holds is equivalent to the weak ℓ^τ norm.

We want to characterize all $\mathscr{A}_q^\alpha(\mathscr{H})$ in terms of coefficients f_k:

We have seen: For any $r > 0$, the nonlinear spaces $\Sigma_n(\mathscr{H})$ satisfy Jackson and Bernstein inequalities for the space $Y := \mathscr{A}_\infty^r(\mathscr{H})$ and

$$\mathscr{A}_q^\alpha(\mathscr{H}) = (\mathscr{H}, \mathscr{A}_\infty^r(\mathscr{H}))_{\alpha/\tau,q}.$$

The mapping $f \to (f_k)$ is invertible and gives an isometry between $\mathscr{H}$ and $\ell^2(\mathbb{N})$ and also between $\mathscr{A}_\infty^r$ and $\ell^{\tau,\infty}(\mathbb{N})$. Interpolation gives that this mapping is also an isometry between $\mathscr{A}_q^\alpha(\mathscr{H})$ and $\ell^{\tau(\alpha),q}(\mathbb{N})$ with τ defined by

$$\frac{1}{\tau} = \alpha + \frac{1}{2}.$$

We have thus the following complete characterization of approximation spaces for *n*-term approximation (see DeVore [1998]):

Theorem 4.10: *For nonlinear n-term approximation in a Hilbert space $\mathscr{H}$, a function f is in $\mathscr{A}_q^\alpha(\mathscr{H})$ if and only if its coefficients are in $\ell^{\tau(\alpha),q}$,*

$$\tau(\alpha) := (\alpha + 1/2)^{-1},$$

and $|f|_{\mathscr{A}_q^\alpha(\mathscr{H})} \asymp \|(f_k)\|_{\ell^{\tau(\alpha),q}}$.

4.2.2 Approximation and wavelets

In this subsection, we consider the relations of approximation theory and wavelets. We begin with characterizations of function spaces via properties of the wavelet coefficients of the functions belonging to these spaces, and show then how these characterizations can be used to study linear and nonlinear approximation with wavelets. We use again DeVore [1998].

In the following, we consider only the separable wavelets in $L^2(\mathbb{R}^d)$ constructed via a univariate scaling function φ and a wavelet ψ. The dilation matrix is thus given as a diagonal $(d \times d)$ matrix $A = \mathrm{diag}(2, \ldots, 2)$, and $M := |\det A| = 2^d$. We define the set D of indices to be

$$D := \left\{ (i,j,k) \;\middle|\; i = \left\{1, \ldots, 2^d - 1\right\}, j \in \mathbb{Z}, k \in \mathbb{Z}^d \right\}$$

and the sets D_j of indices in scale j to be

$$D_j := \left\{ (i,j,k) \;\middle|\; i = \left\{1, \ldots, 2^d - 1\right\}, k \in \mathbb{Z}^d \right\}.$$

For each index $I = (i,j,k) \in D$ we further define

$$|I| := M^j = 2^{dj},$$

i.e. as the size of the characteristic hypercube $[0, 2^j)^d$ of the scale j (our definition of D is slightly different from that given in DeVore [1998]).

Different normalizations It is sometimes convenient to choose normalizations for the wavelets (and hence coefficients) that are different from the normalizations belonging to $L^2(\mathbb{R}^d)$. We define them slightly different from DeVore [1998]. The normalization for $L^p(\mathbb{R}^d)$, $0 < p \leq \infty$ shall be:

$$\psi_{I,p} := |I|^{-1/p+1/2}\psi_I, \qquad I \in D$$

with a similar definition for the dual functions. Thus, with $1/p + 1/p' = 1$, we get

$$f = \sum_{I\in D} c_{I,p}(f)\psi_{I,p}, \qquad c_{I,p} := \langle f, \tilde{\psi}_{I,p'}\rangle.$$

It is easy to go from one normalization to another, e.g., for any $0 < p,q \leq \infty$:

$$\psi_{I,p} = |I|^{1/q-1/p}\psi_{I,q}, \qquad c_{I,p}(f) = |I|^{1/p-1/q}c_{I,q}(f).$$

Characterization of L^p spaces by wavelet coefficients The basis here is the Littlewood-Paley theory of harmonic analysis (we follow again DeVore [1998]). One cannot simply characterize L^p spaces by ℓ^p norms of wavelet coefficients. Rather, one must go through the square function:

$$S(f,x) := \left(\sum_{I\in D} c_{I,2}(f)^2|I|^{-1}\mathbf{1}_{[0,2^j)^d}(x)\right)^{1/2} = \left(\sum_{I\in D} c_{I,p}(f)^2|I|^{-2/p}\mathbf{1}_{[0,2^j)^d}(x)\right)^{1/2}$$

which incorporates the interaction between the scales. For $1 < p < \infty$, one has

$$\|f\|_{L^p(\mathbb{R}^d)} \asymp \|S(f,\cdot)\|_{L^p(\mathbb{R}^d)}$$

with the constants of equivalency only depending on p. This can be extended to $p \leq 1$ if L^p is replaced by the Hardy space H_p (see appendix) and more assumptions are made on the wavelet ψ.

Characterization of Besov spaces by wavelet coefficients Consider Besov spaces $B^\alpha_q(L^p(\mathbb{R}^d))$ for $0 < q,p \leq \infty$, $\alpha > 0$. Then, for all $f \in B^\alpha_q(L^p(\mathbb{R}^d))$, we have (see DeVore [1998]):

$$|f|_{B^\alpha_q(L^p(\mathbb{R}^d))} \asymp \begin{cases} \left(\sum_{j=-\infty}^{\infty} 2^{j\alpha q}\left[\sum_{I\in D_j} c_{I,p}(f)^p\right]^{q/p}\right)^{1/q}, & 0 < q < \infty, \\ \sup_{j\in\mathbb{Z}} 2^{j\alpha}\left[\sum_{I\in D_j} c_{I,p}(f)^p\right]^{1/p}, & q = \infty. \end{cases}$$

We can define spaces of functions for all $\alpha > 0$ by using the right side (finiteness). But these spaces will *coincide* with Besov spaces only for a certain range of α and p depending on the wavelet ψ.

- In the case $1 \leq p \leq \infty$ we need that
  - **(a)** $\psi \in B^\beta_q(L^p(\mathbb{R}^d))$ for some $\beta > \alpha$,
  - **(b)** ψ has r vanishing moments for some $r > \alpha$.

- In the case $p < 1$, we also need that $r > d/p - d$. Then the space $B_q^\alpha(H_p(\mathbb{R}^d))$ is characterized (replace the L^p modulus of smoothness by H_p modulus of smoothness, see Kyriazis [1996]). However, if $\alpha > d/p - d$ this space is the same as $B_q^\alpha(L^p(\mathbb{R}^d))$.

For a fixed value of $1 \le p < \infty$, the spaces $B_\tau^\alpha(L^\tau(\mathbb{R}^d))$, $1/\tau = \alpha/d + 1/p$, occur in nonlinear approximation. If we choose the wavelets normalized in L^p, then the norm equivalence in this case becomes simply:

$$|f|_{B_\tau^\alpha(L^\tau(\mathbb{R}^d))} \asymp \Big(\sum_{I\in D} c_{I,p}(f)^\tau\Big)^{1/\tau}.$$

Shift invariant spaces and linear approximation

We follow DeVore [1998]. Consider the shift invariant spaces $V_j := V_j(\varphi)$ with a scaling function φ for (linear!) approximation in the $L^2(\mathbb{R}^d)$-norm. Let

$$E_j(f) := E_j(f)_2 := \inf_{S\in V_j} \|f - S\|_{L^2(\mathbb{R}^d)}, \qquad j = 0, 1, \ldots.$$

As we mentioned in subsection 4.1.5, the spaces V_j provide approximation if φ satisfies the Strang-Fix conditions. Moreover, it is easy to prove the Jackson inequality: For all f in the Sobolev space $W^r(L^2(\mathbb{R}^d))$, we have

$$E_j(f) \le C2^{-jr}|f|_{W^r(L^2(\mathbb{R}^d))}, \qquad j = 0, 1, \ldots.$$

The companion Bernstein inequality is

$$|S|_{W^r(L^2(\mathbb{R}^d))} \le C2^{jr}\|S\|_{L^2(\mathbb{R}^d)}, \qquad S \in V_j,$$

which is valid if φ is in $W^r(L^2(\mathbb{R}^d))$. Thus: Under these conditions on φ, the general theory applies and we get the following characterization of approximation spaces (the same as for other linear approximations):

$$\mathscr{A}_q^\alpha(L^2(\mathbb{R}^d)) = B_q^\alpha(L^2(\mathbb{R}^d)), \qquad 0 < \alpha < r,\ 0 < q \le \infty.$$

We have a similar theory for approximation in $L^p(\mathbb{R}^d)$, $1 \le p \le \infty$, and even $0 < p < 1$.

Nonlinear wavelet approximation

We use again DeVore [1998]. We consider now *n*-term approximation with wavelets. The intuitive idea is: If the target function is smooth on some region, use a coarse resolution in that region; this amounts to putting terms in the approximation corresponding to coarse scale. On regions were the target function is not smooth we use higher resolution, i.e. take terms from finer scales. Questions arising from this intuitive observations are (see DeVore [1998]):

(i) How should we measure smoothness to make such demarcations between smooth and nonsmooth?

(ii) How do we allocate terms in a nonlinear strategy?

(iii) Are there precise characterizations of functions that can be approximated with a given approximation order?

All of these questions have a definitive and simple solution.

Considered will be only approximation in L^p, $1 < p < \infty$; but more generality is possible: Essential is only the equivalence of function norms with norms on the sequence of wavelet coefficients. Thus, the results hold equally well for Hardy spaces H_p (Cohen et al. [2000]). First, we begin with the case $\mathbb{R}^d$, $d \geq 1$, and then consider the extension of results to other domains.

Let φ, $\tilde{\varphi}$ be two scaling functions which are in duality and let ψ and $\tilde{\psi}$ be their wavelets. Each function $f \in L^p(\mathbb{R})$ has then a wavelet decomposition. Let Σ_n^w denote the set of all functions

$$S = \sum_{I \in \Lambda} a_I \psi_I$$

where $\Lambda \subseteq D$ is a set of indices of cardinality $\#\Lambda \leq n$. Thus, Σ_n^w is the set of all functions which are a linear combination of n wavelet functions. Define:

$$\sigma_n^w(f)_p := \inf_{S \in \Sigma_n^w} \|f - S\|_{L^p(\mathbb{R})}.$$

The characterization of classes for n-term approximation is done by proving the Jackson and Bernstein inequalities. The original proof was by DeVore et al. [1992]; simpler techniques can be found in Cohen et al. [2000].

Jackson and Bernstein inequalities They hold if ψ has sufficient vanishing moments and sufficient smoothness (see DeVore [1998]):

Theorem 1: *Let $1 < p < \infty$, let $s > 0$ and let $f \in B_\tau^s(L^\tau(\mathbb{R}))$, $1/\tau = s + 1/p$. Let ψ satisfy the following conditions:*

(i) *ψ has r vanishing moments with $r > s$*

(ii) *ψ is in $B_q^\rho(L^\tau(\mathbb{R}))$ for some q and some $\rho > s$.*

Then, the Jackson inequality

$$\sigma_n(f)_p \leq C|f|_{B_\tau^s(L^\tau(\mathbb{R}))} n^{-s}, \qquad n = 1, 2, \ldots,$$

holds with C depending only on p and s, and the Bernstein inequality

$$\|f\|_{B_\tau^s(L^\tau(\mathbb{R}))} \leq C n^s \|f\|_{L^p(\mathbb{R})}, \qquad n = 1, 2, \ldots,$$

holds if $f = \sum_{I \in \Lambda} c_{I,p}(f) \psi_{I,p}$ with $\#\Lambda \leq n$.

For the multivariate case $\mathbb{R}^d$, replace $1/\tau = s + 1/p$ by $1/\tau = s/d + 1/p$ and $n^{\pm s}$ by $n^{\pm s/d}$.

Approximation spaces for n-term approximation With this we can characterize the approximation spaces (see DeVore [1998]). Let $1 < p < \infty$ and $s > 0$ and let $1/\tau := s/d + 1/p$. If ψ satisfies the vanishing moments and smoothness conditions needed for the Jackson and Bernstein inequalities, then, for any $0 < \gamma < s$ and any $0 < q \leq \infty$:

$$\mathscr{A}_q^{\gamma/d}(L^p(\mathbb{R}^d)) = \left(L^p(\mathbb{R}^d), B_\tau^s(L^\tau(\mathbb{R}^d))\right)_{\gamma/s,q}.$$

Concerning this equation, we give several remarks (following DeVore [1998]):

- The interpolation spaces on the right side are the approximation spaces for free knot spline approximation and $d = 1$ (in higher dimensions, free knot spline approximation is not understood!).
- There is one value for q where the right side is a Besov space: if $1/q = \gamma/d + 1/p$, the right side is $B_q^\gamma(L^q(\mathbb{R}^d))$ with equivalent norms.
- There is a description of the interpolation spaces on the right side in terms of wavelet coefficients: A function is in the space

  $$\left(L^p(\mathbb{R}^d), B_\tau^s(L^\tau(\mathbb{R}^d))\right)_{\gamma/s,q}$$

  if and only if $(c_{I,p}(f))_{I\in D}$ is in the Lorentz space $\ell^{\mu,q}$ where $1/\mu := \gamma/d + 1/p$ and we have

  $$|f|_{\mathscr{A}_q^{\gamma/d}(L^p(\mathbb{R}^d))} \asymp \|(c_{I,p})(f)\|_{\ell^{\mu,q}}.$$

  (This verifies the previous remark for the case $q = \mu$.)
- For each n, let f_n denote a best n-term approximation to f in $L^p(\mathbb{R}^d)$ (which can be shown to exist, see Temlyakov [1998]); then

  $$K(f, n^{-s}, L^p(\mathbb{R}^d), B_\tau^s(L^\tau(\mathbb{R}^d))) = \|f - f_n\|_{L^p(\mathbb{R}^d)} + n^{-s}|f_n|_{B_\tau^s(L^\tau(\mathbb{R}^d))}$$

  i.e., f_n realizes the K-functional at $t = n^{-s}$.

Concluding, n-term wavelet approximation offers an attractive alternative to free knot spline approximation: In one dimension, the only case where free knot spline approximation is completely understood, it provides the same approximation efficiency and yet is more easily numerically implementable.

Wavelet decompositions and n-term approximations on domains We follow DeVore [1998]. Let $\Omega \subseteq \mathbb{R}^d$ be a Lipschitz domain, i.e. the boundary $\partial\Omega$ of Ω is Lipschitz. Then: Any function f in a Besov space $B_q^\alpha(L^p(\Omega))$ can be extended to all of $\mathbb{R}^d$ in such a way that the extended function Ef satisfies

$$|Ef|_{B_q^\alpha(L^p(\mathbb{R}^d))} \leq C|f|_{B_q^\alpha(L^p(\Omega))}$$

(see DeVore and Sharpley [1984] and DeVore and Sharpley [1993]). The extended function Ef has a wavelet decomposition, and the previous results can be applied. The n-term approximation to Ef will provide the same order of approximation to f on Ω, and one can delete in the approximant all terms corresponding to wavelets that are not active in Ω (i.e. all wavelets whose support does not intersect Ω). The problem is that numerical implementation is not always easy.

Another approach applicable in certain settings is the construction of a wavelet basis for the domain Ω. Particularly suitable is this in the case of an interval $\Omega \subseteq \mathbb{R}$: Biorthogonal wavelets can be constructed for an interval (see Cohen et al. [1993]) and can easily be extended to parallelepipeds in $\mathbb{R}^d$ and even polyhedral domains (see Dahmen [1997]).

n-term approximation: Numerical considerations We make the following assumptions (see DeVore [1998]):

- The approximation takes place in a domain $\Omega \subseteq \mathbb{R}^d$ which admits a biorthogonal basis.
- For simplicity of notation, assume $d = 1$.
- The wavelet decomposition of the target function f is finite and known to us. (If the wavelet decomposition is not finite, one usually assumes more about f that allows truncation of the wavelet series while retaining the desired level of accuracy.)

For best n-term approximation in $L^2(\Omega)$:

- Choose the n terms in the wavelet series of f for which the absolute value of the coefficients is largest.

Generalization to L^p: Write f in its wavelet expansion with respect to L^p-normalized wavelets:

- Choose the n terms in the wavelet series of f for which $|c_{I,p}(f)|$ is largest.

The resulting approximant f_n will provide the Jackson estimate for n-term approximation. It gives also a near best approximant:

$$\|f - f_n\|_{L^p(\Omega)} \leq C\sigma_n(f)_p, \qquad n = 1,2,\ldots,$$

with a constant C independent of f and n (Temlyakov [1998]).

The selection of the largest coefficients seems to make necessary a sorting of the coefficients. But this sorting can be avoided by the use of ***thresholding***: Given a tolerance $\varepsilon > 0$, let Λ_ε be the set of all intervals I for which $|c_{I,p}(f)| > \varepsilon$. Define the ***hard thresholding operator***

$$T_\varepsilon(f) := \sum_{I \in \Lambda_\varepsilon(f)} c_{I,p}(f)\psi_{I,p} = \sum_{|c_{I,p}(f)|>\varepsilon} c_{I,p}(f)\psi_{I,p}.$$

If the target function f is in weak ℓ^τ with $1/\tau = s + 1/p$, then it follows from the definition of this space that

$$\#(\Lambda_\varepsilon(f)) \leq M^\tau \varepsilon^{-\tau}$$

with M the weak ℓ^τ norm of the coefficients, $M := |f|_{\ell^{\tau,\infty}}$. One obtains:

$$\|f - T_\varepsilon(f)\|_{L^p(\Omega)} \leq CM^{\tau/p}\varepsilon^{1-\tau/p}.$$

For example, if $\varepsilon = MN^{-1/\tau}$, then $\#(\Lambda_\varepsilon(f)) \leq N$ and

$$\|f - T_\varepsilon(f)\|_{L^p(\Omega)} \leq CMN^{-1}.$$

Thus: Thresholding provides the Jackson estimate, and therefore provides the same approximation efficiency as n-term approximation.

Let

- $M := |f|_{\ell^{\tau,\infty}}$,
- ε a thresholding tolerance,
- η a prescribed error, and
- N a prescribed number of coefficients.

Then the following table records the relation between thresholding and n-term approximation (taken from DeVore [1998]):

Threshold	Number of coefficients	Error
ε	$M^\tau\varepsilon^{-\tau}$	$M^{\tau/p}\varepsilon^{1-\tau/p}$
$M^{-1/(ps)}\eta^{1/(s\tau)}$	$M^{1/s}\eta^{-1/s}$	η
$MN^{-1/\tau}$	N	MN^{-s}

Hard thresholding has a certain instability: coefficients just below the threshold are set to zero, those just above are kept. This can be remedied by the following modification: Given $\varepsilon > 0$, define

$$s_\varepsilon(x) := \begin{cases} 0, & \text{if } |x| \leq \varepsilon, \\ 2(|x| - \varepsilon)\operatorname{sign} x, & \text{if } \varepsilon \leq |x| \leq 2\varepsilon, \\ x, & \text{if } |x| > 2\varepsilon. \end{cases}$$

Then the operator

$$T'_\varepsilon(f) := \sum_{I \in D} s_\varepsilon(c_{I,p}(f))\psi_{I,p}$$

has the same approximation properties as T_ε.

4.2.3 Highly nonlinear approximation

Some questions concerning n-term approximation arise (DeVore [1998]):

- How does the effectiveness of n-term approximation depend on the wavelet basis?
- Is there any advantage gained by adaptively choosing a basis which depends on the target function f?

In many applications like signal processing or statistical estimation, it is not clear which orthonormal system is to be used best. Generally, a class $\mathscr{L}$ of bases is called ***library***. One example are wavelet packets. We formulate the problem in a Hilbert space $\mathscr{H}$ with a library $\mathscr{L}$ of orthonormal bases: Given a target function $f \in \mathscr{H}$, choose both a basis $B \in \mathscr{L}$ and an n-term approximation to f from this basis. This is a ***highly nonlinear problem***. Sometimes, even non-orthonormal systems have to be taken into account. This leads to general dictionaries where one replaces the library $\mathscr{L}$ of bases by a subset $\mathscr{D} \subseteq \mathscr{H}$ of arbitrary functions. We first consider libraries of orthonormal bases in a Hilbert space.

One example are the wavelet packets. Another example of a wavelet library is given by the following construction (see DeVore [1998]):

Take $\mathscr{H} = L^2(\mathbb{R}^2)$ and consider a compactly supported scaling function $\varphi \in L^2(\mathbb{R})$ with orthonormal shifts and corresponding wavelet ψ. Define:

$$\psi^0 := \varphi \qquad \psi^1 := \psi.$$

To each vertex e of the unit square $[0,1]^2$, each $j = (j_1, j_2) \in \mathbb{Z}^2$, $k = (k_1, k_2) \in \mathbb{Z}^2$, associate the function

$$\psi^e_{j,k}(x_1, x_2) := 2^{(j_1+j_2)/2} \psi^{e_1}(2^{j_1}x_1 - k_1)\psi^{e_2}(2^{j_2}x_2 - k_2)$$

(remark the mixing of levels!). Each of these functions has $L^2(\mathbb{R}^2)$ norm one. Let $\mathscr{L}$ denote the library of all complete orthonormal systems which can be made up from these functions. This $\mathscr{L}$ includes the multivariate wavelet bases build by the tensor product constructions.

A special case is the following: Let $\varphi = \mathbf{1}_{[0,1]}$ and $\psi = H$, the Haar function. Approximate functions on the unit square $\Omega := [0,1)^2$. The library $\mathscr{L}$ includes bases of the following type: Take an arbitrary partition $\mathscr{P}$ of Ω into dyadic rectangles R; on each R we can take a standard or tensor product wavelet Haar basis. This library of bases is closely related to the CART algorithm studied by Donoho [1997], and thus to the partitions given by decision trees.

Adaptive basis selection

We follow DeVore [1998]. Let $B := (\eta_k)$ be an orthonormal basis for $\mathscr{H}$ and let $\Sigma_n(B)$ denote the functions which can be written as a linear combination of n of the functions η_k, $k = 0, 1, \ldots$. Further let

$$\sigma_n(f, B) := \sigma_n(f, B)_{\mathscr{H}} := \inf_{S \in \Sigma_n(B)} \|f - S\|_{\mathscr{H}}$$

be the corresponding approximation error. As seen above the decrease of approximation errors $\sigma_n(f, B)$ is completely determined by the rearranged coefficients $\langle f, \eta_k \rangle$. Let $\gamma_k(f, B)$ be the k-th largest of the absolute values of these coefficients. As seen before, for any $\alpha > 0$, a function f from $\mathscr{H}$ is in $\mathscr{A}^\alpha_\infty$ (i.e. $\sigma_n(f, B) = O(n^{-\alpha})$, $n \to \infty$), if and only if $(\gamma_n(f, B))$ is in weak ℓ^τ (i.e. in $\ell^{\tau,\infty}$) with $\tau := (\alpha + 1/2)^{-1}$. Moreover:

$$\|(\gamma_n(f, B))\|_{\ell^{\tau,\infty}} \asymp |f|_{\mathscr{A}^\alpha_\infty}$$

with constants of equivalency independent of B.

Suppose $\mathscr{L} = (B)_{B\in\mathscr{L}}$ is a library of orthonormal bases. Define the ***approximation error***

$$\sigma_n^{\mathscr{L}}(f)_{\mathscr{H}} := \inf_{B\in\mathscr{L}} \sigma_n(f,B)_{\mathscr{H}}$$

and the approximation classes $\mathscr{A}_q^\alpha(\mathscr{H},\mathscr{L})$ in the ususal way as the set of all functions $f \in X$ such that

$$|f|_{\mathscr{A}_q^\alpha(\mathscr{H},\mathscr{L})} := \begin{cases} \left(\sum_{n=1}^\infty \left[n^\alpha \sigma_n^{\mathscr{L}}(f)_{\mathscr{H}}\right]^q \frac{1}{n}\right)^{1/q}, & 0 < q < \infty, \\ \sup_{n\geq 1} n^\alpha \sigma_n^{\mathscr{L}}(f)_{\mathscr{H}}, & q = \infty, \end{cases}$$

is finite. Few is known about the characterization of the approximation classes. A trivial observation is that we have the upper estimate

$$\sigma_n^{\mathscr{L}}(f)_{\mathscr{H}} \leq Cn^{-\alpha} \inf_{B\in\mathscr{L}} \|(\gamma_n(f,B))\|_{\ell^{\tau,\infty}}$$

with C an absolute constant. Moreover, for any α:

$$\bigcap_B \mathscr{A}_\infty^\alpha(\mathscr{H},B) \subseteq \mathscr{A}_\infty^\alpha(\mathscr{H},\mathscr{L}).$$

DeVore [1998] gives the following interpretation for $q = \infty$, which easily generalizes to any $0 < q \leq \infty$: For each basis B the condition $(\gamma_n(f)) \in \ell^{\tau,\infty}$, $\tau := (\alpha + 1/2)^{-1}$ can be viewed as a smoothness condition relative to the basis B. The infimum on the right side of the inequality characterizing $\sigma_n^{\mathscr{L}}(f)_{\mathscr{H}}$ can be thought of as the infimum of smoothness conditions relative to the different bases B. Similarly, the classes $\mathscr{A}_\infty^\alpha(\mathscr{H},B)$ are smoothness classes with respect to B, and $\bigcap_B \mathscr{A}_\infty^\alpha(\mathscr{H},B)$ is an intersection of smoothness classes. The advantage of optimal basis selection is thus: We are allowed to take the basis $B \in \mathscr{L}$ in which $\mathscr{L}$ is smoothest. The problem is that the above characterizations for $\sigma_n^{\mathscr{L}}(f)_{\mathscr{H}}$ and $\mathscr{A}_\infty^\alpha(\mathscr{H},B)$ cannot be reversed in general. Whether they can be reversed in concrete cases is not known.

Adaptive basis selection for wavelet packets

An example were we have an algorithm for adaptive basis selection is in the case of wavelet packets (see DeVore [1998], but our presentation differs from his).

Let $T = (G,r)$, $G = (V,E)$, be a fully expanded M-ary rooted tree of height $h \in \mathbb{N}$ with the set of associated strings $\mathscr{B}$. Let Γ_b for $b \in \mathscr{B}$ be the spaces given by a wavelet packet decomposition, and let γ_b be the corresponding bases.

As we have seen: n-term approximation efficiency using orthonormal bases is related to ℓ^τ norms of the coefficients. The algorithm in this case is then:

- Fix an integer m for the desired numerical accuracy.
- Choose $\tau > 0$.
- Find a basis for the ℓ^τ norm as described in the following.

Let f be the target function. The coefficients

$$\langle f, \gamma_b \rangle$$

can then be computed efficiently with the wavelet filters $\tilde{H}_i$.

Let $B = (\eta_I)$ be any orthonormal subcollection of basis functions and define

$$N_\tau(B) := N_\tau(f, B) := \sum_B |\langle f, \eta_I \rangle|^\tau.$$

We want to find the basis B which minimizes this.

We do this by going from level to level towards the root, beginning with the highest level. For each node u with associated string b, we choose a basis B_b in the following way: If u is a leaf, we set $B_b = \gamma_b$; if u is an inner node, we have two bases for the space Γ_b:

$$\gamma_b \qquad \text{and} \qquad B_{b0} \cup \cdots \cup B_{b(M-1)}$$

where B_{bi} are the bases chosen in the upper level chosen earlier. We compare

$$N_\tau(\gamma_b) \qquad \text{with} \qquad N_\tau(B_{b0} \cup \cdots \cup B_{b(M-1)})$$

and choose B_b to be the basis which minimizes this. At the root r, we have found the best basis B.

Highly nonlinear approximation: Dictionaries

Following Temlyakov [2002], we define dictionaries for arbitrary Banach spaces X:

Definition 4.7: *Let $\mathscr{X}$ be a Banach space with norm $\|\cdot\|$. We call a set $\mathscr{D} \subseteq \mathscr{X}$ of functions from $\mathscr{X}$ a **dictionary** if each $g \in \mathscr{D}$ has norm one, $\|g\| = 1$, and the closure of* span $\mathscr{D}$ *coincides with $\mathscr{X}$. For simplicity, we assume that with $g \in \mathscr{D}$, we have also $-g \in \mathscr{D}$.*

Dictionaries should be limited to cases which are computationally feasible.

Perhaps the first example with a redundant dictionary was considered by Schmidt [1907]: He approximated functions $f(x,y)$ of two variables by bilinear forms

$$\sum_{i=1}^{m} u_i(x) v_i(y)$$

in $L^2([0,1]^2)$. This problem is closely connected with the properties of the integral operator

$$J_f(g) := \int_0^1 f(x,y) g(y) dy$$

with kernel $f(x,y)$ (see DeVore [1998], Temlyakov [2002]).

Other examples are:

- Neural networks,

- Gabor functions,
- Anharmonic Fourier Analysis.

A common feature to all examples is the ***redundancy***: There are many more functions in the dictionary than needed to approximate any target function f. The hope is that the redundancy will increase the efficiency. But redundancy may also slow down the search for a good approximation. Results on highly nonlinear approximation are quite fragmentary and a cohesive theory still needs to be developed.

Approximation using n-terms from a dictionary

We follow DeVore [1998], Temlyakov [2002] and Barron et al. [to appear]. Suppose that $\mathscr{D}$ is a dictionary from a Hilbert space $\mathscr{H}$. A special case of a dictionary $\mathscr{D}$ is given when $\mathscr{D}$ is an orthonormal basis of $\mathscr{H}$.

For each $n \in \mathbb{N}$, let $\Sigma_n := \Sigma_n(\mathscr{D})$ denote the collection of all functions in $\mathscr{H}$ which can be expressed as a linear combination of at most n elements of $\mathscr{D}$. Then, each function $S \in \Sigma_n$ can be written in the form

$$S = \sum_{g \in \Lambda} c_g g, \qquad \Lambda \subseteq \mathscr{D}, \qquad \#\Lambda \leq n,$$

with the $c_g \in \mathbb{R}$; it may be possible to write S in more than one way.

For a function $f \in \mathscr{H}$, we define its approximation error

$$\sigma_n(f)_{\mathscr{H}} := \sigma_n(f, \mathscr{D})_{\mathscr{H}} := \inf_{S \in \Sigma_n} \|f - S\|_{\mathscr{H}}.$$

Interest lies in an estimate for σ_n (from above and below). For this purpose, introduce the following way to measure smoothness with respect to the dictionary $\mathscr{D}$: For a general dictionary $\mathscr{D}$ and for any $\tau > 0$, define the class of functions

$$\mathscr{K}_\tau^o(\mathscr{D}, M) := \left\{ f \in \mathscr{H} \;\middle|\; f = \sum_{g \in \Lambda} c_g g,\ \Lambda \subseteq \mathscr{D},\ \#\Lambda < \infty \text{ and } \sum_{g \in \Lambda} |c_g|^\tau \leq M^\tau \right\},$$

and define $\mathscr{K}_\tau(\mathscr{D}, M)$ as the closure (in $\mathscr{H}$) of $\mathscr{K}_\tau^o(\mathscr{D}, M)$.

Furthermore, define $\mathscr{K}_\tau(\mathscr{D})$ as the union of the classes $\mathscr{K}_\tau(\mathscr{D}, M)$ over all $M > 0$ and the semi-norm

$$|f|_{\mathscr{K}_\tau(\mathscr{D})}$$

as the infimum of all M such that $f \in \mathscr{K}_\tau(\mathscr{D}, M)$. When $\tau = 1$, then $\mathscr{K}_1$ is the class of functions which are a convex combination of functions in $\mathscr{D}$.

In the case where $\mathscr{D}$ is a basis B, n-term approximation from $\mathscr{D}$ is the same as n-term approximation from B. We have seen that if $1/\tau = \alpha + 1/2$, then f is in the approximation class $\mathscr{A}_\tau^\alpha(\mathscr{D})$ if and only if

$$\sum_k |\langle f, h_k \rangle|^\tau$$

is finite and this expression is equivalent to $|f|^\tau_{\mathscr{A}_\tau(B)}$. In particular, this shows that

$$\sigma_n(f, \mathscr{D})_{\mathscr{H}} \leq C n^{-\alpha} |f|_{\mathscr{K}_\tau(\mathscr{D})}$$

in the special case that $\mathscr{D}$ is given by an orthonormal basis B.

There is an interest in understanding whether this holds for more general dictionaries $\mathscr{D}$.

- For the case $\alpha = 1/2$, this result is due to Maurey (see Pisier [1980]), who showed that the above inequality is valid for any dictionary. An iterative algorithm to generate approximants from $\Sigma_n(\mathscr{D})$ that achieves this estimate for $\alpha = 1/2$ was given by Jones [1992].
- For $\alpha \geq 1/2$, the estimate is proved in DeVore and Temlyakov [1996].
- For $\alpha < 1/2$ ($1 \leq \tau \leq 2$), there seems to be no obvious analogue for general dictionaries.

Greedy algorithms

In the following we use DeVore [1998], Temlyakov [2002] and Barron et al. [to appear]. Greedy algorithms are also known as ***adaptive pursuit***, ***matching pursuit*** in signal processing, or ***projection pursuit*** in the neural networks literature. Since best m-term approximations are usually out of reach, greedy algorithms aim at building suboptimal but good m-term approximations. We will mention three variants of greedy algorithms.

The Pure Greedy Algorithm (PGA) The first algorithm is the pure greedy algorithm. It can be applied for any dictionary $\mathscr{D}$ and has the advantage of simplicity. It begins with a target function $f \in \mathscr{H}$ and successively generates approximants

$$G_m(f) \in \Sigma_m(\mathscr{D}) \qquad m = 1, 2 \ldots .$$

In the case that $\mathscr{D}$ is generated by an orthonormal basis B, $G_m(f)$ is a best m-term approximation to f.

If $f \in \mathscr{H}$, let $g = g(f) \in \mathscr{D}$ denote an element from $\mathscr{D}$ which maximizes $\langle f, g \rangle$:

$$\langle f, g(f) \rangle = \sup_{g \in \mathscr{D}} \langle f, g \rangle .$$

Assume for simplicity that such maximizer exists; if not, suitable modifications are necessary in the algorithm that follows.

Define:

$$G(f) := G(f, \mathscr{D}) := \langle f, g(f) \rangle g(f)$$

and

$$R(f) := R(f, \mathscr{D}) := f - G(f).$$

That means, $G(f)$ is the best one-term approximation to f from $\mathscr{D}$ and $R(f)$ is the residual of this approximation.

Then the ***pure greedy algorithm (PGA)*** is (DeVore [1998], Temlyakov [2002], Barron et al. [to appear]):

- Initially, set $R_0(f) := R_0(f, \mathscr{D}) := f$ and $G_0(f) := G_0(f, \mathscr{D}) := 0$.

- For each $m \geq 1$, inductively define

$$G_m(f) := G_m(f, \mathscr{D}) := G_{m-1}(f) + G(R_{m-1}(f)),$$
$$R_m(f) := R_m(f, \mathscr{D}) := f - G_m(f) = R(R_{m-1}(f)).$$

The pure greedy algorithm converges to f for each $f \in \mathscr{H}$ (see Davis et al. [1997]). It is greedy in the sense that at each iteration it approximates the residual $R_m(f)$ as best possible by a single function from $\mathscr{D}$. If $\mathscr{D}$ is generated by an orthogonal basis, then it is easy to see that $G_m(f)$ is a best m-term approximation to f from $\mathscr{D}$ and

$$\sigma_m(f, B)_{\mathscr{H}} = \|f - G_m(f)\|_{\mathscr{H}} = \|R_m(f)\|_{\mathscr{H}}.$$

For general dictionaries $\mathscr{D}$ this is not the case. Approximation properties of this algorithm are far from being optimal:

DeVore and Temlyakov [1996] showed the following estimate to hold: For each $f \in \mathscr{K}_1(\mathscr{D})$:

$$\|f - G_m(f)\|_{\mathscr{H}} \leq |f|_{\mathscr{K}_1(\mathscr{D})} m^{-1/6},$$

which could slightly be improved to

$$\|f - G_m(f)\|_{\mathscr{H}} \leq 4|f|_{\mathscr{K}_1(\mathscr{D})} m^{-11/62}$$

in Konyagin and Temlyakov [1999]. Moreover, there is an example of a dictionary $\mathscr{D}$ and a function f which is a linear combination of two elements of $\mathscr{D}$ such that

$$\|f - G_m(f)\|_{\mathscr{H}} \geq C m^{-1/2}$$

with C an absolute constant. This means that for the simplest functions (which are in all smoothness classes $\mathscr{K}_\tau(\mathscr{D})$), the pure greedy algorithm provides approximation of at most order $O(m^{-1/2})$. Livshitz and Temlyakov [2003] could show that there exist a dictionary $\mathscr{D}$ and an element $f \in \mathscr{H}$, $f \neq 0$, with an even lower bound:

$$\|f - G_m(f)\|_{\mathscr{H}} \geq C m^{-0.27}.$$

This means that the PGA cannot provide estimates

$$\sigma_n(f, \mathscr{D})_{\mathscr{H}} \leq C n^{-\alpha} |f|_{\mathscr{K}_\tau(\mathscr{D})}$$

for $\alpha > 0.27$.

We proceed with some modifications of the pure greedy algorithm with more favourable approximation properties (DeVore [1998], Temlyakov [2002], Barron et al. [to appear]).

The Relaxed Greedy Algorithm (RGA)

The ***relaxed greedy algorithm (RGA)*** is:

- Define $R_0^r(f) := R_0^r(f, \mathscr{D}) := f$ and $G_0^r(f) := G_0^r(f, \mathscr{D}) := 0$.

- For $m = 1$, define

$$G_1^r(f) := G_1^r(f, \mathscr{D}) := G_1(f),$$
$$R_1^r(f) := R_1^r(f, \mathscr{D}) := R_1(f).$$

- For each $m \geq 2$, inductively define

$$G_m^r(f) := G_m^r(f, \mathscr{D}) := \Big(1 - \frac{1}{m}\Big) G_{m-1}^r(f) + \frac{1}{m} g(R_{m-1}(f)),$$
$$R_m^r(f) := R_m^r(f, \mathscr{D}) := f - G_m^r(f),$$

  where, as before, for a function $h \in \mathscr{H}$, let $g = g(h)$ denote a function from $\mathscr{D}$ which maximizes $\langle h, g \rangle$.

Thus, the relaxed greedy algorithm is less greedy than the pure greedy algorithm: it uses the ***relaxation parameter*** $1/m$. Jones [1992] showed that the relaxed greedy algorithm provides approximation order

$$\| f - G_m^r(f) \|_{\mathscr{H}} \leq C m^{-1/2}, \qquad m = 1, 2, \ldots,$$

for any $f \in \mathscr{K}_1(\mathscr{D})$. Unfortunately, this estimate requires the knowledge that $f \in \mathscr{K}_1(\mathscr{D})$; if this information is not available (as would be the case in most applications) the choice of the relaxation parameter as $1/m$ is not appropriate.

The Orthogonal Greedy Algorithm (OGA) Another variant, called orthogonal greedy algorithm, removes some of the objections to the choice of the relaxation parameter.

Let us shortly return to the pure greedy algorithm. As seen above, it chooses functions

$$g_j := G(R_j(f)), \qquad j = 1, \ldots, m.$$

It does not provide the best approximation from the span of $g_1, \ldots, g_m$. If H_0 is a finite-dimensional subspace of H, let P_{H_0} be the orthogonal projector from H onto H_0, i.e. $P_{H_0}(f)$ is the best approximation to f from H_0.

The ***orthogonal greedy algorithm (OGA)*** is (DeVore [1998], Temlyakov [2002], Barron et al. [to appear]):

- Define $R_0^o(f) := R_0^o(f, \mathscr{D}) := f$ and $G_0^o(f) := G_0^o(f, \mathscr{D}) := 0$.

- For each $m \geq 1$, inductively define

$$H_m := H_m(f) := \operatorname{span}\big\{ g(R_0^o(f)), \ldots, g(R_{m-1}^o(f)) \big\}$$

  and

$$G_m^o(f) := G_m^o(f, \mathscr{D}) := P_{H_m}(f),$$
$$R_m^o(f) := R_m^o(f, \mathscr{D}) := f - G_m^o(f).$$

Thus, the orthogonal greedy algorithm takes the best approximation by linear combinations of the functions

$$G(R_0(f)),\ldots,G(R_{m-1}(f))$$

available at each iteration. If the dictionary $\mathscr{D}$ is an orthonormal basis, then PGA and OGA coincide. DeVore and Temlyakov [1996] have shown that the orthogonal greedy algorithm satisfies the estimate

$$\|f - G_m^o(f,\mathscr{D})\|_{\mathscr{H}} \leq |f|_{\mathscr{K}_1(\mathscr{D})} m^{-1/2}.$$

From this, it is easy to derive (DeVore and Temlyakov [1996]):

Theorem 4.11: *Let $\mathscr{D}$ be any dictionary, let $\alpha \geq 1/2$ and $1/\tau = \alpha + 1/2$; if $f \in \mathscr{K}_\tau(\mathscr{D})$, then*

$$\sigma_m(f,\mathscr{D})_{\mathscr{H}} \leq C|f|_{\mathscr{K}_\tau(\mathscr{D})} m^{-\alpha}, \qquad m = 1,2,\ldots.$$

For OGA and RGA, Barron et al. [to appear] could provide convergence rates $m^{-\alpha}$, $0 < \alpha < 1/2$, whenever f belongs to a certain intermediate space between $\mathscr{K}_1(\mathscr{D})$ and the Hilbert space $\mathscr{H}$, namely the spaces

$$\mathscr{B}_p := [\mathscr{H}, \mathscr{K}_1(\mathscr{D})]_{\theta,\infty}, \quad \theta := 2/p - 1, \quad 1 < p < 2,$$

which are the real interpolation spaces between $\mathscr{H}$ and $\mathscr{K}_1(\mathscr{D})$. They showed that if $f \in \mathscr{B}_p$, then the OGA and RGA, when applied to f, provide approximation rates $Cm^{-\alpha}$ with

$$\alpha := \theta/2 = 1/p - 1/2.$$

Thus, if one sets $\mathscr{B}_1 = \mathscr{K}_1(\mathscr{D})$, then these spaces provide a full range of approximation rates for greedy algorithms. The results are optimal in the sense that one recovers the best possible convergence rate in the case where the dictionary is an orthonormal basis.

Other greedy algorithms It should be noted that the described greedy algorithms are not ready for implementation. The term "algorithm" is actually not really justified, we are concerned only with algorithm schemes. Indeed, the implementation may result difficult because the search for the best choice in each step may not be easy. With neural networks, it really is not easy (see e.g. Hush [1999]).

To relax these problems, weak versions of all greedy algorithms, called Weak Greedy Algorithms, have been developed, which are more apt for implementation. Here, in each step only a nearly best solution has to be found in a certain neighbourhood of the best solution. The bounds of these neighbourhoods then have to be tightened as m goes to infinity. See Temlyakov [2002] for further details.

For greedy algorithms in general Banach spaces see also Temlyakov [2002].

4.3 Wavelets and Bayesian techniques: Denoising

We will now describe an important application where the combination of Bayesian and multiresolution methods have achieved an ernormous success: The removel of noise from a real-world signal or image. We follow presentations found in Figueiredo and Nowak [2001] and Abramovich et al. [1998].

Non-parametric regression models Suppose Y is a noisy signal or image, modelled as a stochastic process

$$Y = X + V$$

where X denotes the process describing the undisturbed signal and V is the noise process which is usually assumed to be white and Gaussian.

Usually, X is given by a function $g(t)$ (where t means time or space), and only disturbed observations $y = (y_1, \ldots, y_n)^\top$ on equidistant time or space points t_i are known. For simplicity, we may assume $t_i \in \mathbb{Z}^d$. Then the standard ***non-parametric regression problem*** is:

$$y_i = g(t_i) + \varepsilon_i, \qquad i = 1, \ldots, n,$$

where $t_i \in \mathbb{Z}$, ε_i are i.i.d. with

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2).$$

By defining $x := (x_1, \ldots, x_n)^\top$ with $x_i := g(t_i)$ and $\varepsilon := (\varepsilon_1, \ldots, \varepsilon_n)^\top$, we could as well write

$$y = x + \varepsilon, \qquad \varepsilon \sim \mathcal{N}_n(0, \sigma^2 I)$$

where I denotes the identity matrix of appropriate size. We assume that the variance σ^2 is known; otherwise, it has to be estimated, e.g. by the MAD (Median Absolute Deviation) algorithm, see Donoho and Johnstone [1995].

The task is then to recover the unknown function g from noisy data y_i without assuming any parametric form for g.

General possibilities to solve this task are:

- spline smoothing,
- kernel estimation,
- generalized Fourier series expansion,
- wavelets.

Basic properties of DWT for signal processing Wavelets and other multiscale analysis tools are used successfully in signal/image processing for the following tasks (see e.g. Figueiredo and Nowak [2001]):

- approximation/representation,
- estimation,
- compression.

In all these applications, two important properties of the discrete wavelet transform (DWT) of real-world signals and images are exploited:

1) The coefficients tend to be much less correlated than the original data.

2) The representation in wavelet coefficients is ***sparse***, i.e. a few large coefficients dominate this representation.

The decorrelation of the wavelet coefficients is a result of the multiresolution analysis as given in section 4.1. The sparseness property is justified by the properties of nonlinear *n*-term approximation, the connections to smoothness spaces and the corresponding decay conditions of the coefficient spaces presented in section 4.2. These both properties together with the fast implementation of Mallat's FWT make DWT an excellent tool for signal processing.

The basic approach to DWT-based signal processing consists in manipulating the DWT coefficients rather than the signal samples themselves. DWT-based signal/image processing follows a three step program (see Figueiredo and Nowak [2001]):

1) Compute the DWT coefficients of the signal.

2) Perform some specified processing on these coefficients.

3) Compute the inverse DWT to obtain the processed signal.

In the denoising context, the decorrelation property suggests processing the coefficients independently of each other; the sparseness property ("heavy-tailedness") paves the way to the use of threshold/shrinkage estimators.

Discrete Wavelet Transform and Gaussian errors The vector $\omega := (c_I)$ of ***sample discrete wavelet coefficients*** is given by performing the discrete wavelet transform (DWT):

$$\omega = \mathscr{W} y.$$

The vector $\theta := (c_I^*)$ of ***population discrete wavelet coefficients*** is defined as the DWT of the function values $x := (g(t_1), \ldots, g(t_n))$:

$$\theta = \mathscr{W} x.$$

Applying the DWT $\mathscr{W}$ to the noisy data y leads to noisy coefficients ω

$$\omega := \mathscr{W} y = \mathscr{W} x + \mathscr{W} \varepsilon = \theta + \varepsilon'$$

where

$$\theta := \mathscr{W} x$$

and

$$\varepsilon' =: \mathscr{W} \varepsilon \sim \mathscr{N}_n(0, \sigma^2 I),$$

since $\mathscr{W}$ is orthogonal, i.e. $\mathscr{W}\mathscr{W}^\top = \mathrm{Id}$. We have thus for each coefficient

$$c_I = c_I^* + \varepsilon_I'$$

with the sample coefficients $\omega = (c_I)$, the population coefficients $\theta = (c_I^*)$ and the noise $\varepsilon' = (\varepsilon_I')$, where $I \in D$ denote appropriate indices for the wavelet coefficients (as described in subsection 4.1.5).

The next step is to extract those coefficients that contain information about the unknown vector x, and discard the others. This can be done by thresholding the sample discrete wavelet coefficients c_I. The intuitive idea is: The true vector x has a parsimonious wavelet expansion, i.e. only a few "large" c_I essentially contain information about x. One has to decide which ones these are and to set the others to zero.

The general denoising procedure with thresholding is (see e.g. Abramovich et al. [1998]):

1) Expand noisy data y_i in wavelet series.

2) Extract "significant" wavelet coefficients by thresholding.

3) Invert wavelet transform for denoised coefficients.

Thresholding is actually a special case of shrinkage (Abramovich et al. [1998]):

Shrinkage rule:

- Decreases (not necessarily strictly) the absolute values of the wavelet coefficients without changing their sign.

Thresholding rule:

- Additionally: Maps to zero all coefficients falling in some non-empty interval around zero.

Wavelet estimators with properly chosen threshold rule have various important optimality properties (Donoho and Johnstone [1994], Donoho et al. [1995]).

Thresholding rules

Let c_I be an arbitrary DWT coefficient of the observed signal/image. Then the ***hard and soft thresholding estimators*** (Donoho and Johnstone [1994], Donoho and Johnstone [1995]) are defined as (Figueiredo and Nowak [2001])

$$\delta_\lambda^{\text{hard}}(c_I) := c_I \mathbf{1}_{\lambda,\infty}(|c_I|) = \begin{cases} 0, & \text{if } |c_I| \le \lambda, \\ c_I, & \text{if } |c_I| > \lambda, \end{cases}$$

$$\delta_\lambda^{\text{soft}}(c_I) := \text{sign}(c_I)\max(0, |c_I| - \lambda) = \begin{cases} 0, & \text{if } |c_I| \le \lambda, \\ \text{sign}(c_I)(|c_I| - \lambda), & \text{if } |c_I| > \lambda. \end{cases}$$

Soft thresholding yields systematically biased estimators, because it shrinks coefficients regardless of how large they are. Hard thresholding produces less biased but higher variance estimates.

The ***nonnegative garrote*** (Gao [1998]) tries to retain the best of both approaches (see figure 4.5):

$$\delta_\lambda^{\text{garrote}}(c_I) = \begin{cases} 0, & \text{if } |c_I| \le \lambda \\ c_I - \frac{\lambda^2}{c_I}, & \text{if } |c_I| > \lambda. \end{cases}$$

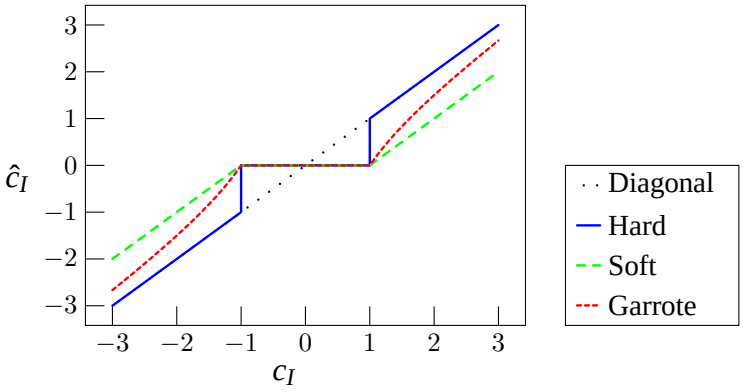


Figure 4.5: Hard, soft and garrote thresholding function

Define $\hat{c}_I := \delta_\lambda(c_I)$ where δ_λ is one of the thresholding rules, and construct an estimator $\hat{x}$ of x by applying the inverse DWT:

$$\hat{x} = \mathscr{W}^\top \hat{\theta} \qquad \text{where } \hat{\theta} := (\hat{c}_I).$$

The general problem is how to choose the threshold for the thresholding rule. Proposed thresholds λ are (Abramovich et al. [1998]):

- *VisuShrink* (Donoho and Johnstone [1994]): The ***universal threshold***

$$\lambda_{\mathrm{DJ}} = \sigma\sqrt{2\log(n)}.$$

  The resulting nonlinear wavelet estimator is spatially adaptive and asymptotically near-minimax within the whole range of Besov spaces. Moreover, it outperforms any linear estimator (i.e. splines, kernel estimation, truncated Fourier series, etc.) within Besov spaces $B^s_{p,q}$ with $p < 2$ that contain spatially inhomogeneous functions. However, it depends on the data only through the estimated σ and thus oversmoothes in practice.

Therefore, data-driven thresholding rules have been proposed:

- *SureShrink* (Donoho and Johnstone [1995]): Based on minimizing Stein's unbiased risk estimate (Stein [1981]); yields usually smaller thresholds than *VisuShrink*. Asymptotically near-minimax, overall complexity $O(n\log(n))$.

- Cross-validaton (Nason [1995], Nason [1996]; Weyrich and Warhola [1995]).

- Multiple Hypothesis Testing (Abramovich and Benjamini [1995], Abramovich and Benjamini [1996]; Ogden and Parzen [1996b], Ogden and Parzen [1996a]).

- Bayesian viewpoint (introduced by Vidakovic [1998], Clyde et al. [1998], Chipman et al. [1997]).

The most thresholding procedures are essentially minimax and thus too conservative. They do not take into account some specific properties of a concrete vector x or function g. The natural way of introducing prior belief (knowledge, information) about g (e.g. regularity properties) is via a Bayesian approach: Specify a prior distribution on the population wavelet coefficients c_I^*.

The approach presented here is within a Bayesian framework:

- Impose a prior on the wavelet coefficients of the unknown response function.
- Estimate the function by applying some Bayes rule on the resulting posterior distribution of the wavelet coefficients.

The main goal is to design the prior model as to capture the sparseness of the wavelet expansion common to most applications.

Bayesian formulation

The likelihood function resulting from the observation model in the signal domain is given by

$$y \,|\, x \sim \mathcal{N}_n(x, \sigma^2 I),$$

and in the wavelet domain by

$$\omega \,|\, \theta \sim \mathcal{N}_n(\theta, \sigma^2 I).$$

Noise is white and Gaussian both in signal and wavelet domain.

The prior $\pi_\Theta(\theta)$ is formulated with respect to the wavelet coefficients. This prior induces a signal prior

$$\pi_{\mathcal{X}}(x) = \pi_\Theta(\mathcal{W} x),$$

because $\mathcal{W}$ is an orthogonal transformation, thus possesses a unit Jacobian, i.e.

$$|d\theta| = |dx|.$$

The Bayesian version of the three step program for wavelet-based denoising is (Figueiredo and Nowak [2001])

1) Compute the DWT of the data $\omega = \mathcal{W} y$;

2) Obtain a Bayes estimate $\hat{\theta}$ given ω;

3) Reconstruct the signal estimate $\hat{x} = \mathcal{W}^{-1}\hat{\theta}$.

Let $L(\theta, \tilde{\theta})$ be a loss function which penalizes the "discrepancy" between θ and any candidate $\tilde{\theta}$ and define (see e.g. Figueiredo and Nowak [2001])

$$\hat{\theta} := \arg\min_{\tilde{\theta}} \int L(\theta, \tilde{\theta}) \pi(\theta \,|\, \omega) d\theta.$$

Then

$$\hat{x} := \mathcal{W}^{-1} \arg\min_{\tilde{\theta}} \int L(\theta, \tilde{\theta}) \pi(\theta \,|\, \omega) d\theta$$

which is equivalent to

$$\hat{x} := \arg\min_{\tilde{x}} \int L(\mathscr{W}x, \mathscr{W}\tilde{x})\pi(x|y)dx$$

since

$$\pi(x|y) \propto \pi(y|x)\pi_{\mathscr{X}}(x) = \pi(\omega|\theta)\pi_{\mathscr{X}}(\mathscr{W}^{-1}\theta) = \pi(\omega|\theta)\pi_{\Theta}(\theta) \propto \pi(\theta|\omega).$$

The estimate $\hat{x} = \mathscr{W}^{-1}\hat{\theta}$ corresponds to a Bayesian criterion in the signal domain under the loss

$$L(\mathscr{W}x, \mathscr{W}\tilde{x})$$

induced by $L(\theta, \tilde{\theta})$. In some cases, this loss is ***invariant under orthogonal transformations***, in the sense that:

$$L(\mathscr{W}x, \mathscr{W}\tilde{x}) \propto L(x, \tilde{x}).$$

In this case:

$$\hat{x} := \arg\min_{\tilde{x}} \int L(x, \tilde{x})\pi(x|y)dx,$$

meaning that $\hat{x} = \mathscr{W}\hat{\theta}$ is a Bayes estimator under the same loss function as $\hat{\theta}$.

Examples of invariant loss functions are (Figueiredo and Nowak [2001]):

- Squared error loss L_2. Then, the optimal Bayes rule is the posterior mean (PM):

  $$L_2(\theta, \tilde{\theta}) = \|\theta - \tilde{\theta}\|_2^2 = \|\mathscr{W}x - \mathscr{W}\tilde{x}\|_2^2 = \|\mathscr{W}(x - \tilde{x})\|_2^2 = \|x - \tilde{x}\|_2^2 = L_2(x, \tilde{x})$$

  because of the orthogonality of $\mathscr{W}$. Thus, the inverse DWT of the PM estimate of coefficients coincides with the PM estimate of x.

- 0-1 loss $L_{0\text{-}1}$: Leads to the maximum a posteriori (MAP) criterion

  $$L_{0\text{-}1}(\theta, \tilde{\theta}) = L_{0\text{-}1}(\mathscr{W}x, \mathscr{W}\tilde{x}) = L_{0\text{-}1}(x, \tilde{x})$$

  simply because $\mathscr{W}^{-1}$ exists, i.e. because $\mathscr{W}$ is bijective. Thus, the inverse DWT of the MAP estimate of the coefficients is the MAP estimate of x.

This is not true in general! An example is the following:

$$L_\infty(\theta, \tilde{\theta}) = \|\mathscr{W}x - \mathscr{W}\tilde{x}\|_\infty \neq \|x - \tilde{x}\|_\infty = L_\infty(x, \tilde{x})$$

where $\|v\|_\infty := \max\{|v_i|\}$ denotes the infinity norm.

The Bayes rule based on the L_2-loss (posterior mean) leads to a shrinkage rule rather than a thresholding rule (Vidakovic [1998], Clyde et al. [1998], Chipman et al. [1997]). In contrast, a Bayes rule based on a weighted combination of L_1-losses (posterior median), which corresponds to L_1-losses based on the function g and its derivatives yields for certain priors a thresholding rule (see for example Abramovich et al. [1998]).

Priors

The decorrelation property suggests to model the coefficients as mutually independent (though of course decorrelation does not imply independence! We take it as an approximation):

$$\pi(\theta) = \prod_{I \in D} \pi(c_I^*).$$

Since furthermore the likelihood function is conditionally independent, the posterior distribution is as well conditionally independent:

$$\pi(\theta \,|\, \omega) \propto \prod_{I \in D} \pi(c_I \,|\, c_I^*) \prod_{I \in D} \pi(c_I^*) \propto \prod_{I \in D} \pi(c_I^* \,|\, c_I)$$

where $\pi(c_I^* \,|\, c_I) \propto \pi(c_I \,|\, c_I^*)\pi(c_I^*)$ with $c_I \,|\, c_I^* \sim \mathcal{N}(c_I^*, \sigma^2)$.

Under the MAP or the PM criterion, the Bayes rule can be computed separately for each coefficient (Figueiredo and Nowak [2001]):

$$\hat{\theta}_{\mathrm{PM}} = \mathbf{E}[\theta \,|\, \omega] = \big(\mathbf{E}[c_I^* \,|\, c_I]\big)_{I \in D}$$
$$\hat{\theta}_{\mathrm{MAP}} = \arg\max_{\theta} \pi(\theta \,|\, \omega) = \big(\arg\max_{c_I^*} \pi(c_I^* \,|\, c_I)\big)_{I \in D}.$$

Focus now on the prior for *one* wavelet coefficient. The usual approach is to explicitly capture the sparseness property with heavy-tailed priors; examples are (Figueiredo and Nowak [2001]):

- Chipman et al. [1997], Crouse et al. [1998]: mixture of two zero-mean Gaussian, one with small variance, other with large variance.
- Abramovich et al. [1998]: as before, but small variance component as point mass at zero.
- Vidakovic [1998]: Student t-distributions.
- Laplacian prior: $\pi(c_I^*) \propto \exp(-\nu|c_I^*|)$ and MAP rule leads to soft thresholding function.
- Bayesian interpretation of hard thresholding rule was presented by Moulin and Liu [1999].

Choices of prior In [Chipman and Wolfson, 1999], a comparison is made between several choices of priors for the wavelet coefficients. Actually, a look on the non-parametric model

$$y = \mathscr{W}\theta + \varepsilon$$

and

$$\omega = (c_I) = \mathscr{W}^{-1} y, \qquad \omega \sim \mathcal{N}_n(\theta, \sigma^2 I)$$

reveals the following dependence structure (graphical model):

$$\begin{array}{ccc} & & \theta \\ & & | \\ \sigma^2 & — & \omega \\ & & | \\ \mathscr{W} & — & y \end{array}$$

Thus, priors have to be put on the "parameters" $\mathscr{W}$, θ, and σ^2.

- $\mathscr{W}$: Priors on $\mathscr{W}$ are seldom considered; the wavelet basis used is usually taken to be fixed. Possibilities for a choice are
  - wavelet families (e.g. Daubechies family),
  - member of the family (e.g. number of filter coefficients),
  - basis of a wavelet packet.

  Whereas finding a prior for the choice of a wavelet family seems to be very difficult, a prior on the member of a given family is more approachable because (as in the case of Daubechies wavelets) the members are often indexed by integers N corresponding to the smoothness properties of the wavelets. The last point, choice of a wavelet packet basis, would result in a Bayesian analogue of the basis selection algorithm presented in subsection 4.2.3. In any of these cases, the reversible jump sampler (see subsection 3.5.1) could be usable when treating the problem with a Bayesian simulation approach.

- σ^2: A classical choice for a prior on σ^2 is an inverse Gamma distribution

  $$\sigma^2 \sim \mathscr{IG}(\alpha/2, \beta/2).$$

  Special cases are $\sigma^2 = s^2$ fixed ($\alpha \to \infty$, $\beta \to \infty$, $\alpha/\beta = 1/s^2$), or an uninformative prior given by $\alpha = \beta = 0$.

- θ: The priors on the unobserved wavelet coefficients θ are most varying in the literature. There are three main possibilities:
  - $\theta \sim \mathscr{N}_n(0, \Sigma)$: The covariance matrix Σ expresses the dependencies between the wavelet coefficients. Often it is chosen to be dependent on the variance σ:

    $$\Sigma = \sigma^2 \Sigma',$$

    where the choice $\Sigma' = \mathbf{I}$ means independence of the coefficients in θ. Another choice of Σ (or Σ') is some kind of "band structure", such that coefficients with strong prior correlation would be those which are "close" to each other, i.e. coefficients at similar locations, similar scales, or corresponding locations in similar scales. (Also a Student's t-distribution can be used instead of a normal distribution.)

– Mixture of two normal distributions:

$$c_I^* = \pi_I \mathcal{N}(0, \tau_I^2) + (1-\pi_I)\mathcal{N}(0, \rho_I^2), \qquad c_I^* \text{ independent of each other,}$$

with $\tau_I^2 \gg \rho_I^2$ and $\pi_I \in [0,1]$. The small variance ρ_I^2 belongs to the "negligible" coefficients, the large variance τ_I to the "significant" coefficients: the standard deviation ρ_I is related to the largest possible coefficient which may be shrunk towards zero, whereas the standard deviation τ_I is related to the largest possible "significant" coefficient. The limiting case $\rho_I^2 = 0$ leads to a mixture of a normal and a delta distribution on 0:

$$c_I^* = \pi_I \mathcal{N}(0, \tau_I^2) + (1-\pi_I)\delta_0, \qquad c_I^* \text{ independent of each other.}$$

While a choice $\rho_I^2 > 0$ always leads to a shrinkage rule, with $\rho_I^2 = 0$ a thresholding rule is possible (see the next section). In applications where a compression (i.e. reduction of non-zero wavelet coefficients) is important, this latter choice is to be preferred. Through further prior modelling on the weights π_I also dependencies on similar locations and scales can be considered.

– Infinite mixture of normal distributions:

$$\theta \sim \mathcal{N}_n(0, \Sigma)$$

with hyperparameter on the covariance matrix Σ (Holmes and Denison [1999]). Classical hyperparameters like inverse Gamma or Wishart distributions help little to the task of prior elicitation, because prior knowledge lies more in the complexity and underlying smoothness of the signal rather than in the values of the parameters themselves. Considering the case where $\Sigma = \mathrm{diag}(v_I)$ is diagonal, a natural measure for model complexity is given by the degrees of freedom

$$D_F = (1 + \sigma^2 v_I^{-1})^{-1}$$

(see Hastie and Tibshirani [1990]). A preference for smooth models with low degrees of freedom is naturally expressed by the prior

$$v_I^{-1} \,|\, \sigma^2 \propto \exp\big(-c(1+\sigma^2 v_I^{-1})^{-1}\big)$$

where the constant c determines how much to penalize model complexity. The log of the posterior for this prior is found to be

$$\text{Log Model Probability} = \text{Log Marginal Likelihood} - c \times D_F$$

showing the form of many classical model choice criteria. This in turn allows to choose the hyperparameter c according to these criteria:

c	0	1	$\frac{1}{2}\log n$	$\log n$
Model choice criterion	Bayes factor	AIC	BIC	RIC

An example of a prior

We present now the prior given by Abramovich et al. [1998]. Using a mixture of a normal and a delta distribution and using a median as estimator leads to a thresholding rule. This choice reveals also an interesting connection between certain hyperparameters and some Besov space parameters.

As already mentioned, a large variety of different functions allow parsimonious representation in wavelet series: Only a few non-negligible coefficients are present in the expansion. One possibility is to incorporate this by placing the following prior on c_I^* (Abramovich et al. [1998]):

$$c_I^* \sim \pi_I \mathcal{N}(0, \tau_I^2) + (1 - \pi_I)\delta_0$$

with $0 \leq \pi_I \leq 1$, δ_0 the point mass at zero, and c_I^* independent of each other. The hyperparameters π_I, τ_I^2 have to be specified.

This prior is a mixture of a point mass at zero and a normal distribution around zero: Every c_I^* is either zero with probability $1 - \pi_I$ or, with probability π_I, normally distributed with zero mean and variance τ_I^2. The probability π_I gives the proportion of non-zero wavelet coefficients while the variance τ_I^2 is a measure of their magnitudes.

Conjugate posterior distribution The proposed prior is conjugate for the regression model with Gaussian noise. Thus, the posterior distribution for $c_I^* \,|\, c_I$ is also a mixture of a normal distribution and a point mass δ_0. Defining

$$\bar{\lambda}_I := \tau_I^2/(\sigma^2 + \tau_I^2),$$

the posterior cumulative function $F(c_I^* \,|\, c_I)$ results to be:

$$F(c_I^* \,|\, c_I) = \frac{1}{1+w_I}\Phi\left(\frac{c_I^* - c_I\bar{\lambda}_I}{\sigma\sqrt{\bar{\lambda}_I}}\right) + \frac{w_I}{1+w_I}\mathbf{1}_{[0,+\infty)}(c_I^*)$$

where Φ is the normal cumulative function and w_I is the ***posterior odds ratio*** for the component at zero, given by:

$$w_I = \frac{1-\pi_I}{\pi_I}\frac{\tau_I}{\sigma\sqrt{\bar{\lambda}_I}}\exp\left(-\frac{\bar{\lambda}_I c_I^2}{2\sigma^2}\right).$$

Usage of L_1 loss As mentioned, the L_2-loss leads to the posterior mean as corresponding Bayes rule, as used by Vidakovic [1998], Clyde et al. [1998], Chipman et al. [1997]. This in turn leads to a shrinkage rule, not a thresholding rule.

Instead, one may use any weighted combination of L_1-losses on the individual wavelet coefficients. The corresponding Bayes rule results in taking the posterior median of each wavelet coefficient. This leads to a thresholding rule.

The ***posterior median*** $\mathrm{Med}(c_I^* \,|\, c_I)$ is defined as the solution (in c_I^*) of the equation

$$F(c_I^* \,|\, c_I) = 1/2.$$

The posterior cumulative distribution function has a jump at zero. Thus: $\mathrm{Med}(c_I^* \,|\, c_I) = 0$ if

$$w_I \geq 1$$

or

$$w_I < 1 \quad \text{and} \quad \frac{1}{2}(1-w_I) \leq \Phi\left(-\frac{\sqrt{\bar{\lambda}_I}c_I}{\sigma}\right) \leq \frac{1}{2}(1+w_I)$$

and $\mathrm{Med}(c_I^* \,|\, c_I) \neq 0$ otherwise. Straightforward calculus leads to

$$\mathrm{Med}(c_I^* \,|\, c_I) = \mathrm{sign}(c_I)\max(0,\zeta_I)$$

where

$$\zeta_I = \bar{\lambda}_I|c_I| - \sqrt{\bar{\lambda}_I}\sigma\Phi^{-1}\left(\frac{1+\min(w_I,1)}{2}\right).$$

The quantity ζ_I is negative for all c_I in some implicitly defined interval

$$[-\lambda_I, \lambda_I].$$

Thus, the estimate $\hat{c}_I$ is zero whenever $|c_I|$ falls below the threshold λ_I. The posterior median in this case is a coefficient dependent thresholding rule with thresholds λ_I. For large c_I the thresholding rule asymptotes to a linear shrinkage rule with factor

$$\bar{\lambda}_I = \tau_I^2/(\sigma^2+\tau_I^2)$$

since the second term in the equation for ζ_I becomes negligible for $|c_I| \to \infty$.

Particular form of the hyperparameters The hyperparameters π_I and τ_I^2 have to be defined. A possible choice with interesting relations to Besov space parameters was proposed by Abramovich et al. [1998] for the one-dimensional case ($d = 1$). They choose level dependent hyperparameters π_j and τ_j^2 for each level j with

$$\pi_j = \pi_I \qquad \text{and} \qquad \tau_j^2 = \tau_I \qquad \text{for all } I \in D_j$$

in the following way:

$$\tau_j^2 = C_1 2^{-\alpha j} \quad \text{and} \quad \pi_j = \min(1, C_2 2^{-\beta j}), \quad j = 0,\ldots,J-1$$

where C_1, C_2, α, β are nonnegative constants.

Remark: The universal threshold $\lambda_{\mathrm{DJ}} = \sigma\sqrt{2\log(n)}$ can be obtained as a particular limiting case (Abramovich et al. [1998]):

$$\alpha = \beta = 0, \quad C_1 \longrightarrow \infty, \quad C_2 \longrightarrow \infty$$

such that

$$\sqrt{C_1}/(C_2\sigma n) \longrightarrow 1.$$

Interpretation of hyperparameters α and β The prior expected number of non-zero wavelet coefficients on the j-th level is $C_2 2^{j(1-\beta)}$.

- In the case $\beta > 1$, the number of non-zero coefficients in the wavelet expansion is finite (this follows from the first Borel-Cantelli lemma). The Prior model implies thus that g is exactly expressed as a finite wavelet expansion.
- More interesting is the case $0 \leq \beta \leq 1$:
  - The case $\beta = 0$: Corresponds to the prior belief that all coefficients on all levels have the same probability of being non-zero. This characterizes self-similar processes as white noise or Brownian motion, the overall regularity depending on α.
  - The case $\beta = 1$: Assumes that the expected number of non-zero coefficients is non-zero on each level. This is typical e.g. for piecewise polynomial functions.

Relation between Besov space parameters and hyperparameters of prior model There is an interesting connection between the hyperparameters of the prior and the Besov space parameters established by Abramovich et al. [1998]. We begin with the two-parameter prior given above, which mainly characterizes the primary Besov parameters. Thereafter, a three-parameter version of the prior will also take the parameter q into account.

Theorem 4.12 (Abramovich et al. [1998])**:** *Let ψ be a mother wavelet of regularity r, where*

$$\max(0, 1/p - 1/2) < s < r, \qquad 1 \leq p, q \leq \infty,$$

and let the wavelet coefficients c_I^* of a function g obey the prior model given above with

$$\tau_j^2 = C_1 2^{-\alpha j} \qquad \textit{and} \qquad \pi_j = \min(1, C_2 2^{-\beta j})$$

where $C_1, C_2, \alpha \geq 0$ and $0 \leq \beta \leq 1$. Then $g \in B_{p,q}^s$ almost surely if and only if either

$$s + 1/2 - \beta/p - \alpha/2 < 0$$

or

$$s + 1/2 - \beta/p - \alpha/2 = 0 \qquad \textit{and} \qquad 0 \leq \beta < 1,\ 1 \leq p < \infty,\ q = \infty.$$

If $\beta > 1$, then the number of non-zero coefficients in the wavelet expansion is finite almost surely. Thus, with probability one, g will belong to the same Besov spaces as the mother wavelet ψ, i.e. those for which

$$\max(0, 1/p - 1/2) < s < r, \qquad 1 \leq p, q \leq \infty.$$

With a three parameter prior we can take into account the Besov space parameter q: Introduce a third parameter γ with $-\infty < \gamma < \infty$:

$$\tau_j^2 = C_1 2^{-\alpha j} j^{\gamma}.$$

Then the previous prior is a special case of this prior with $\gamma = 0$.

Theorem 4.13 (Abramovich et al. [1998])**:** *Let ψ be a mother wavelet of regularity r, where*

$$\max(0, 1/p - 1/2) < s < r, \qquad 1 \le p, q \le \infty,$$

and let the wavelet coefficients c_I^* of a function g obey the prior model given above with

$$\tau_j^2 = C_1 2^{-\alpha j} j^{\gamma} \qquad \text{and} \qquad \pi_j = \min(1, C_2 2^{-\beta j})$$

where $C_1, C_2, \alpha \ge 0$, $0 \le \beta \le 1$ and $\gamma \in \mathbb{R}$. Then $g \in B^s_{p,q}$ almost surely if and only if either

$$s + 1/2 - \beta/p - \alpha/2 < 0$$

or

$$s + 1/2 - \beta/p - \alpha/2 = 0 \quad \text{and} \quad \gamma \text{ satisfies the following conditions:}$$

	$p, q < \infty$	$p = \infty$, $q < \infty$	$p < \infty$, $q = \infty$	$p, q = \infty$
$0 \le \beta < 1$	$\gamma < -2/q$	$\gamma < -1 - 2/q$	$\gamma \le 0$	$\gamma \le -1$
$\beta = 1$	$\gamma < -2/q$		$\gamma < 0$	

With the prior used here, the several coefficients c_I are a-priori independent from each other. If dependency is introduced, the relation to Besov spaces is lost. This can be seen through the so-called "shuffle-invariance": a shuffling of the wavelet coefficients belonging to the same level j leads to the same Besov spaces. This cannot be true for dependent priors.

4.4 Wavelets and dynamical systems

4.4.1 Nonparametric estimation

System identification and wavelets The usage of wavelets for system identification is an astonishingly seldom appearing combination in the literature. For example, Hasiewicz [2001] uses wavelets for the identification of Hammerstein models. This model type is a simple form of nonlinear modelling, done by a serial connection of a nonlinear static and a linear dynamical model. Hasiewicz realizes the static part by a wavelet approximation whereas the linear dynamical part is treated with the usual identification procedures of linear systems theory. The input sequence for the identification process has to be stationary.

Another example is Guo et al. [2004]. They use a stochastic state space model where the transition map is represented by a *linear* wavelet approximation. In chapter 5 we will use a similar idea, which in contrast is principally able to exploit the sparseness properties of the wavelet coefficients revealed by *nonlinear* n-term approximation (augmented by a possible incorporation of wavelet packet trees and a Preisach part in the model).

4.4.2 Linear systems and frames

We have introduced wavelet transform, multiresolution analysis, and orthogonal bases for the analysis of functions. There are interesting analogues for linear time-invariant (LTI) systems,

recently developed by Feuer et al. [2006]. The transform used there is a generalization of the known Laplace, Laguerre and Kautz transform, and also generalizes the more recent Hambo transform developed by Heuberger et al. [2003]. Among other things, these transforms are used to the purpose of model reduction. The idea of sparseness in relation with the analysis of functions and the idea of reduction in relation with models are similar. Thus, it may be expected that with this Unified Transform (UT) a sparse representation of linear systems can be achieved, even more because Feuer et al. [2006] show that this Unified Transform yields some kind of multiresolution analysis for LTI systems. Concerning our case of nonlinear systems, it is especially interesting when we look at Local Model Networks with linear local models. If one uses a wavelet decomposition for the weight functions, one could also try using a decomposition of the local models with the Unified Transform.

The model reduction provided by the Unified Transform can be seen as an optimal pole placement in the Laplace domain.

5 Putting things together: Implementation and application

This chapter describes primarily the concrete implementation of the foregoing more theoretical issues, but provides also the missing links between the single building blocks. These building blocks we have to combine are wavelet techniques, state space modelling of differential and hysteretic behaviour of nonlinear systems, and identification of parameters by stochastic techniques, mainly Bayesian techniques in combination with Sequential Monte Carlo (SMC) methods.

One crucial point in our model is the combination of SMC methods and wavelet-based nonlinear approximation. The sparseness properties of the wavelet coefficients and their practical application using thresholding are here essential. In Guo et al. [2004], also a method combining SMC methods and wavelet approximation is proposed. But although they claim that they use wavelet thresholding (they write "wavelet shrinkage"), they compute instead the vector of wavelet coefficients and then "truncate it by keeping only the first κ elements". But this is not thresholding, which would lead to nonlinear approximation. It is rather solely *linear* approximation, and the only parameters they have to estimate is the number κ of coefficients and their values. In our case, the procedure is more involved: We have to estimate *which* of the coefficients we have to keep, and to estimate their values. We will realize this by using appropriate priors on the wavelet coefficients. For the implementation, we also have to provide a data structure (called wavelet tree) which stores only the non-zero wavelet coefficients in an effective way. Nothing of this is done in the mentioned article. Additionally, we are able to deal with hysteretic behaviour.

Overview At the beginning we shortly summarize the crucial points of the last three chapters. A section with three parts is dedicated to our method (and so following Samarskii): one part describing the whole stochastic model, the second part devoted to the identification algorithm including the definition of our main data structure, the wavelet tree, and the third part summarizing the implementation. We end this chapter providing some examples, and a real application: the identification of data taken from measurements of a shock absorber.

Contributions

- Combination of SMC methods and nonlinear wavelet approximation methods.
- Additional flexibility through the possibility to use wavelet packet trees.
- Inclusion of difference and hysteretical properties into the state space model and nonparametric estimation.

- Description and implementation of a new identification algorithm.
- Application of the algorithm to some examples and on a real data set.

5.1 Summary

We shortly summarize those points of the theories reproduced in the foregoing chapters which are of importance for understanding of the model and algorithm proposed in the present chapter. From wavelet theory:

- Sparseness property,
- Decorrelation property.

From Bayesian probability and statistical decision theory:

- Prior distributions on coefficients,
- Sequential Monte Carlo Methods.

Non-parametric approach: Grey-box model as approximation We follow a non-parametric approach. The task here is to approximate a real system Σ by a best model in the set $\mathscr{M}_n$:

$$\begin{array}{ccccccc} \mathscr{M}_0 & \subset & \mathscr{M}_1 & \subset & \mathscr{M}_2 & \subset & \cdots \\ \cup & & \cup & & \cup & & \\ \Sigma_0 & & \Sigma_1 & & \Sigma_2 & \cdots & \longrightarrow \Sigma \end{array}$$

An analogy is function approximation: The task in this case is to approximate a function f by a best function in the set X_n:

$$\begin{array}{ccccccc} X_0 & \subset & X_1 & \subset & X_2 & \subset & \cdots \\ \cup & & \cup & & \cup & & \\ f_0 & & f_1 & & f_2 & \cdots & \longrightarrow f \end{array}$$

The principal idea in both cases is atomic decompositions: Decompose an object Σ into a weighted superposition of simpler objects Σ^ω, $\omega \in \Omega$:

$$\Sigma = \int_{\omega\in\Omega} \Sigma^\omega \mu_\Sigma(\omega)$$

where

- Ω is some index set, and
- μ_Σ is a (signed, complex) measure dependig on Σ.

Increase of „freedom“	↑	Highly Nonlinear Approximation Nonlinear *n*-term Approximation Linear Approximation	↑	Need of more „knowledge“

Table 5.1: Approximation as an inverse problem

As example let $f \in \mathscr{H}$ be a Hilbert space, and $(\eta_i)_{i\in\mathbb{N}}$, an orthonormal basis. Then

$$f = \sum_{i\in\mathbb{N}} f_i \eta_i, \qquad f_i \in \mathbb{R},$$

with approximations

$$f_n := \sum_{i=0}^{n-1} f_i \eta_i \qquad \text{(Linear Approximation)}$$

or

$$f_n := \sum_{i=0}^{n-1} f_{\tau(i)} \eta_{\tau(i)} \qquad \text{(Nonlinear } n\text{-term Approximation)}$$

where in the latter case $\tau : \mathbb{N} \longrightarrow \mathbb{N}$ is a bijection such that

$$|f_{\tau(0)}| \geq |f_{\tau(1)}| \geq |f_{\tau(2)}| \cdots \qquad \text{(rearranged coefficients)}.$$

A special case is given when the orthonormal basis (η_i) is a wavelet basis.

The next step of “nonlinearity” in approximation is highly nonlinear approximation: Choose a basis (η_i) depending on f (best basis search). A prominent example here are wavelet packets.

The search for best approximation (= identification) is always an inverse problem (see table 5.1). This means: Information contained in experimental data is not enough; using only experimental data leads to unstable identification (too much depending on noise).

To avoid difficulties with inverse problems, one uses regularization. The regularization principle is: Use prior knowledge to stabilize the identifications. Usual assumptions are smoothness properties of the function f. In the special case of wavelets, smoothness properties of the function f are given in terms of Besov spaces. These in turn correspond to sparseness properties of the wavelet coefficients c_I, measured in the norm of sequence spaces. Approximation for functions of these spaces can be done via thresholding of the wavelet coefficients.

Why wavelets? A problem in our case is: f can be multi-dimensional. For n-term approximation, there are two major possibilites:

	1-dim	multi-dim
Wavelets	equivalent	easy extension of 1-dim case
Free-Knot Splines		**?**

There are at least two possibilities to use wavelets for the purpose of identification of nonlinear dynamical systems:

Approximation of systems $\Longleftrightarrow$ Approximation of functions

System Σ	Input/Output-Operator Γ	„Characteristic function“
Nonlinear differentiable control system	$y(t) = \eta(x(t))$ $x(t)$ solution of $\dot{x}(t) = f(u,x,t)$	f multi-dimensional
Preisach Hysteresis	$y(t) = \int\limits_{\alpha<\beta} (\Gamma^{\alpha,\beta}u)(t)\mu(\alpha,\beta)$	$f(\alpha,\beta) := \int\limits_{T(\alpha,\beta)} d\mu(\tilde{\alpha},\tilde{\beta})$

$T(\alpha,\beta)$ = Triangle (α,α)–(α,β)–(β,β)

Table 5.2: Interplay between approximation of dynamical systems and function approximation

- It is possible to transform the input u and output y of the dynamical system with the Wavelet Transform and identify the transformed system; but beware: the Wavelet Transform is linear, so the transformed system is necessarily still nonlinear, and there is no reason to believe that the identification of the transformed system is easier than the original system. We will not follow this approach.

- One can approximate the state transition function and/or the output function with adaptive approximation (and with the help of wavelet bases). This is our choice.

Approximation of dynamical systems How can we use function approximation for the approximation of systems? The key observation here is that important dynamical systems can be completely determined by some kind of “characteristic function”; it is then enough to approximate this function. A summary of the interplay between the approximation of dynamical systems and function approximation shows table 5.2

How to find approximations on f? The idea is to estimate wavelet coefficients. We use a Bayesian approach: Put a prior distribution on the wavelet coefficients. Two important properties of the wavelet coefficients give a general guideline on how to do this:

- Decorrelation property of wavelet decomposition: Choose an independent prior for each wavelet coefficient.

- Sparseness of wavelet coefficients: Use heavy-tailed priors.

A possible prior is a mixture of a Dirac and a Gaussian distribution around 0, as e.g. used for denoising of images. This together with an L_1 loss results in a thresholding rule and realizes thus the approximations.

Wavelets used We use Daubechies wavelets because they bear the following properties:

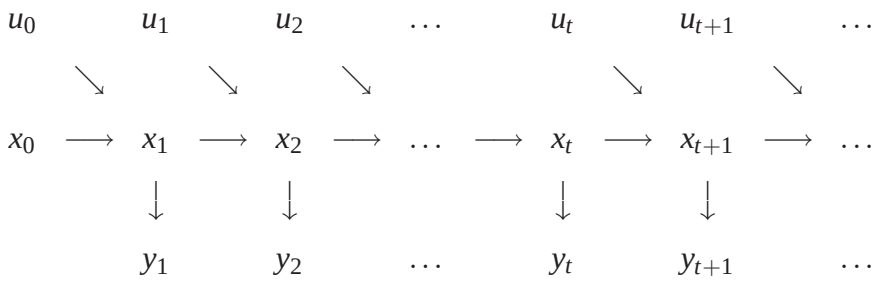
- They have compact support,
- they can be arbitrarily smooth, and
- Fast Wavelet Transform (FWT) is applicable.

Discrete stochastic state space systems with control We consider only discrete state space systems, given by:

$$\text{Controls: } u_t \in \mathbb{R}^n, \qquad \text{States (hidden): } x_t \in \mathbb{R}^d, \qquad \text{Observations: } y_t \in \mathbb{R}^m,$$

with time $t = 0, 1, 2 \ldots$, and

- initial State: $x_0 \sim f_0(x)$,
- state transition equation: $x_t \sim f(x \,|\, x_{t-1}, u_{t-1})$,
- observation equation: $y_t \sim g(y \,|\, x_t)$.



Estimation of states and parameters Estimation of state densities given observed data (filter and smoothing densities): Analytical solutions exist only in a few cases:

- Linear Gaussian Systems $\Longrightarrow$ Kalman-Filter/Smoother,
- Discrete (finite) Systems $\Longrightarrow$ Baum-Welch algorithm.

In all other cases: Exact analytical solutions are not possible. Therefore use the following approximations: Approximate the filter density by a mixture of Dirac distributions and transfer these as particles recursively through the state space system; this leads to Sequential Monte Carlo (SMC) methods (particle filters). The estimation of states is necessary for the estimation of parameters. With the Bayesian viewpoint, there is no difference between states and parameters, and estimation of both can be done jointly with the same methods. The parameters in our case are the wavelet coefficients and possibly other real parameters.

5.2 Model, algorithm and implementation

In this section, we describe our new algorithm together with the underlying model; it follows a short subsection about its implementation.

5.2.1 Model

Assumptions on the system/model

We want the following assumptions on the system to be fulfilled:

- We assume that the system has both differential and hysteretic properties.
- We assume that we can describe the behaviour of the system through a virtual collection of internal signals, given as hidden states consisting of
  - real numbers, and
  - sequences of alternating real numbers of arbitrary length.
- We assume that all information about the differential parts of the model is provided by a finite number of real-valued states.
- We assume that all information about the hysteretic parts of the model is provided by the alternating sequences.
- We assume that the state transitions are provided by either
  - real-valued multi-dimensional functions on some of the real sequences;
  - update of the alternating sequences;
  - summation formulas using a primitive Preisach function on one of the alternating sequences.
- We assume that there are some white noise sources which are subsumed under the real states.
- We assume that there are real inputs (controls) into the system which we subsume under the real states.
- We assume that there is a real output which may be given by a multidimensional real function on the states (including noise sources).

We have thus several real-valued multi-dimensional functions as core of the state transitions. Our aim is the identification of these functions.

The model

The model will be given as a general state space model:

$$\begin{aligned} x_0 &\sim f_0(x) && \text{(initial state)},\\ x_t &\sim f(x\,|\,x_{t-1}, u_{t-1}) && \text{(state transition)},\\ y_t &\sim g(y\,|\,x_t) && \text{(observation)}, \end{aligned}$$

for $t = 1,2,3\ldots$. We first have to fix the spaces wherein to act. The states are taken out of the cartesian product

$$\mathscr{X} := \mathbb{R}^d \times \overline{\mathscr{S}}^e$$

where $\overline{\mathscr{S}}$ is the space of alternating sequences over $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$ and d and e are some natural numbers denoting the respective dimensions.

Building blocks It is possible to factor the state transition $f(x_t \,|\, x_{t-1}, u_{t-1})$ into several, say K, building blocks without disturbing the Markov property: Let $x_{t,1} := (x_t, u_t, \varepsilon_{t,1})$, where x_t is the state at time t, u_t is the control (input) at time t and $\varepsilon_{t,1}$ is some independent noise with known distribution. Define recursively:

$$x_{t,k+1} := a_k(x_{t,k}, u_t, \varepsilon_{t,k}), \qquad k = 1, \ldots, K$$

where $\varepsilon_{t,k}$ are independent noises with known distributions,

$$a_k : \mathbb{R}^{d_k} \times \overline{\mathscr{S}}^{e_k} \to \mathbb{R}^{d_{k+1}} \times \overline{\mathscr{S}}^{e_{k+1}}, \qquad k = 1, \ldots, K-1,$$

are deterministic functions, and d_k and e_k, $k = 1, \ldots, K$ are suitable dimensions. Additionally we require $x_{t+1} := x_{t,K}$. We remark that the requirement that the distributions of the noises are knwon is not really restricting: Parameterized distributions with unknown parameters can usually be easily described as deterministic transformations of standard distributions, for example $\varepsilon \sim \mathscr{N}(\mu, \sigma^2)$ is given by $\varepsilon = \phi(\tilde{\varepsilon}; \mu, \sigma^2)$ with $\tilde{\varepsilon} \sim \mathscr{N}(0,1)$ and the parameterized function

$$\phi(x; \mu, \sigma^2) := \sigma x + \mu.$$

Nevertheless, the noise sources $\varepsilon_{t,k}$ are not restricted to Gaussian noise. Also heavy-tailed distributions are possible. The proposed algorithm for the identification is not restricted to some special assumptions concerning these distributions.

Transitions between the building blocks We will allow the following transitions a_k for the building blocks: Divide each intermediate state $x_{t,k}$ into the part

$$x_{t,k}^{\mathbb{R}} = \left(x_{t,k,1}^{\mathbb{R}}, \ldots, x_{t,k,d_k}^{\mathbb{R}}\right)$$

which gathers all real values, and the part

$$x_{t,k}^{\overline{\mathscr{S}}} = \left(x_{t,k,1}^{\overline{\mathscr{S}}}, \ldots, x_{t,k,e_k}^{\overline{\mathscr{S}}}\right)$$

which gathers all prefixed alternating sequences. Then a_k may realize transitions in three ways, where the first possibility is given by a usual multi-dimensional function and the last two realize a general Preisach operator on some internal one-dimensional signals:

- Real transitions:
$$x_{t,k+1,J}^{\mathbb{R}} := a_{k,j}(x_{t,k,I}^{\mathbb{R}})$$
for some index sets $J \subseteq \{1, \ldots, d_k\}$ and $I \subseteq \{1, \ldots, d_{k+1}\}$.

- Update of alternating sequence:

$$x^{\overline{\mathscr{S}}}_{t,k+1,j}(s) := \rho(x^{\overline{\mathscr{S}}}_{t,k,i_1}, x^{\mathbb{R}}_{t,k,i_2})$$

  for some $j \in \{1,\dots,e_k\}$, $i_1 \in \{1,\dots,e_{k+1}\}$ and $i_2 \in \{1,\dots,d_{k+1}\}$, where ρ denotes the update algorithm of prefixed alternating sequences (see subsection 2.2.2).

- Summation formula:

$$x^{\mathbb{R}}_{t,k+1,j} := \sum_s F_{k,j}\left(x^{\overline{\mathscr{S}}}_{t,k,i}(s), x^{\overline{\mathscr{S}}}_{t,k,i}(s+1)\right)$$

  for some $j \in \{1,\dots,d_k\}$ and $i \in \{1,\dots,e_{k+1}\}$, and where $F_{k,j}$ is a suitable primitive function of a general Preisach hysteresis operator.

Additionally, we allow:

- Sampling from a noise source:

$$x^{\mathbb{R}}_{t,k+1,j} \sim \text{noise source}$$

  for some $j \in \{1,\dots,d_k\}$.

We also have the observation transition:

$$y := b(x^{\mathbb{R}}, \eta_t)$$

for $x^{\mathbb{R}} := x^{\mathbb{R}}_{t,K}$, given by a deterministic function b and independent observation noise η_t.

Parameters We thus are concerned with several multi-dimensional functions: $a_{k,j}$ for the real state transitions, $F_{k,j}$ for the Preiach transitions, and b for the observation transition. For either of them, we assume that they are either known, or, if they are not known, that they are given

- either as parameterized function with an unknown vector θ of real parameters,
- or nonparametric through unknown wavelet coefficients.

Since in the last case we want to use techniques closely related to nonlinear approximation and since we want that these methods provide efficient representations of these functions, we have to assume some smoothness or regularity conditions on the functions. They could e.g. be assumed to be in some suitable Besov space. In this case, the wavelet coefficients can be equipped with independent priors.

Priors for wavelet coefficients and parameters For the wavelet coefficients, we use a prior as described in section 4.3. This prior is a mixture of a delta distribution and a normal distribution, both with mean zero:

$$c^*_I \sim \pi_I \mathcal{N}(0, \tau_I^2) + (1-\pi_I)\delta_0$$

with the weight π_I and variances τ_I^2 only depending on the level j.

For the parameters, the priors are application dependent; the algorithm allows a free choice.

5.2.2 Algorithm

The aim of the algorithm is the identification of the unknown functions in the state and observation transitions. We treat these functions all in the same way: Either they are parameterized, or we approximate them by a wavelet decomposition. We thus have to identify the parameters and the wavelet coefficients. Having identified in some way the wavelet coefficients, we afterwards reconstruct the functions with the usual quadrature mirror filters given by the (inverse) fast wavelet transform. The advantage of this procedure lies in the sparseness and decorrelation properties of the wavelet coefficients: We need by far less data to represent the original function than by storing the values of the function point by point, and the coefficients can be estimated independently: If we change one coefficient, the resulting reconstructed function will change only locally.

The identification of the parameters and the wavelet coefficients has to be done jointly with the hidden states. We use a Bayesian approach and include the parameters and wavelet coefficients into the state space. We then use SMC techniques for the joint estimation of parameters, coefficients and states.

General algorithm Our basic algorithm is the SISR algorithm where resampling is done if the effective sample size falls below some threshold, as described in section 3.6.4. For simplicity, we always use the state transition density as proposal. We have to augment the SISR algorithm with a procedure to evaluate the functions given by wavelet coefficients. We therefore need some special data structures which we will describe in the following.

Storing of the wavelet coefficients To store the wavelet coefficients, we need two kinds of information when using wavelet package bases:

- The wavelet packet tree storing the rules for the reconstruction of the original function.
- The (nonzero) wavelet coefficients themselves.

The storage of the wavelet coefficients can be done similarly to the storage of the entries of a sparse matrix, by providing both location and value only of the nonzero coefficients in a list. During reconstruction of the original function, only these nonzero values have to be used, saving computation time.

Introduction of artificial dynamics for parameters and wavelet coefficients Due to the difficulties SMC methods still have with identifying mixed dynamic and static parameters, we have to modify the model and make the static parameters and wavelet coefficients dynamic: we introduce artificial dynamics (“jitter”). For both parameters and wavelet coefficients, we use some kind of random walk. For the parameters, this is standard. For the wavelet coefficients, we allow a random walk as well as a possibility to threshold a coefficient to zero, with greater possibility if it is small. We also allow the creation of a new wavelet coefficient with some possibility; the value of this new wavelet coefficient is normally distributed. There is no possibility to assess good parameters for these distributions in advance. They depend on the application, and we decided in each case by try and error. In any case, a random walk

consisting of a mixture of two normal distributions (one for small and one for large steps) seems sensible.

Wavelet trees

Wavelet trees are the core data structure of our algorithm. They consist of

- a regular rooted tree, and
- wavelet coefficients attached to each node of this tree.

Wavelet trees realize a multi-dimensional discrete function $h : \mathbb{Z}^d \longrightarrow \mathbb{R}$. The regularity index M depends on the dimensionality d; if tensor product wavelets are used, this index is $M = 2^d$. Each node has then none or exactly $M = 2^d$ children.

We assign coefficients to each node of the tree, both to inner nodes and leaves. The coefficients on the nodes cannot be chosen independently. They are related through the levels via the scaling and wavelet filters. We have the inverse operations of decomposition and reconstruction:

- Decomposition
  - Given the coefficients of a node, the coefficients of its children are obtained by decomposing these coefficients with the scaling and wavelet filters (FWT).
  - Beginning with the root, one computes the coefficients for the next lower level with the filters given by the FWT.
  - Recursively iterating this procedure ends up with the leaves: the original function on the root is thus decomposed into wavelet coefficients.
- Reconstruction
  - Given the coefficients of the children of a node, the coefficients of this node are obtained by reconstructing these coefficients with the inverse scaling and wavelet filters (Inverse FWT).
  - Beginning with the leaves, one computes the coefficients for the next upper level with the filters given by the Inverse FWT.
  - Recursively iterating this procedure ends up with the root: here the coefficients correspond to the values of the realized function.

Definition 5.1: *Let A be a dilation matrix with $M := |\det A|$. Let $T := (G, r)$, $G := (V, E)$, be a rooted M-ary regular tree with a strict enumeration q. Let $\psi_0 := \varphi$ be a scaling function and $\psi_1, \ldots, \psi_M$ be wavelets with associated filter coefficients $(h_{i,m})_{m\in\mathbb{Z}}$, $i = 0, \ldots, M-1$. Let further be $\tilde{\psi}_0 := \tilde{\varphi}$ be a dual scaling function and $\tilde{\psi}_1, \ldots, \tilde{\psi}_M$ the dual wavelets, and let $(\tilde{h}_{i,m})_{m\in\mathbb{Z}}$, $i = 0, \ldots, M-1$, be the dual filters coefficients. Define the decomposition operator*

$$\tilde{H}_i \eta := \sum_{m\in\mathbb{Z}} \tilde{h}_{i,m} \eta(A \cdot -k)$$

and the reconstruction operator

$$H[\eta_0,\ldots,\eta_{M-1}] := \sum_{i=0}^{M-1} \sum_{m\in\mathbb{Z}} h_{i,m}\eta_i(A\cdot -k)$$

as for wavelet packets. A family of coefficients

$$\mathscr{C} := \left\{(c_{u,z})_{u\in V, z\in\mathbb{Z}^d}, c_{u,z}\in\mathbb{R}\right\}$$

is called an ***associated family of wavelet coefficients*** *if for each $u\in V$ and the set of its children* $\mathrm{ch}(u)$ *the following holds:*

- *The coefficients of the (enumerated) children $\{v_0,\ldots,v_{M-1}\} = \mathrm{ch}(u)$ are obtained by decomposition from the coefficients of u,*

  $$(c_{v_i,z})_{z\in\mathbb{Z}^d} = \tilde{H}_i(c_{u,z})_{z\in\mathbb{Z}^d},$$

  or, equivalently,

- *the coefficients of u are obtained by reconstruction from the coefficients of the (enumerated) children $\{v_0,\ldots,v_{M-1}\} = \mathrm{ch}(u)$,*

  $$(c_{u,z})_{z\in\mathbb{Z}^d} = H[(c_{v_0,z})_{z\in\mathbb{Z}^d},\ldots,(c_{v_{M-1},z})_{z\in\mathbb{Z}^d}].$$

The pair $(T,\mathscr{C})$ is then called a ***wavelet tree****.*

Given a wavelet tree $(T,\mathscr{C})$, $\mathscr{C} := \left\{(c_{u,z})_{u\in V,z\in\mathbb{Z}^d}\right\}$, each coefficient is determined by

- the node $u\in V$,
- the key (index) $z\in\mathbb{Z}^d$, and
- its value $c_{u,z}$.

The wavelet trees have the same structure as the wavelet packet trees presented in subsection 4.1.6. But we do not assign the wavelet packet bases to each node; instead we assign coefficients. Both kinds of trees belong together: to each wavelet packet tree those wavelet trees belong with the same structure and where the assigned coefficients decompose and reconstruct according to the filters given by the wavelet packet tree.

Definition 5.2: *Let $(T,\mathscr{C})$ be a wavelet tree with $T := (G,r)$, $G := (V,E)$ and*

$$\mathscr{C} := \left\{(c_{u,z})_{u\in V,z\in\mathbb{Z}^d}\right\}.$$

Let $S := (G_S,r_S)$, $G_S := (U_S,E_S)$, be a regular subtree of T with regularity index equal to that of T. Let further

$$\mathscr{C}(S) := \left\{(c_{u,z})_{u\in V_S,z\in\mathbb{Z}^d}\right\}.$$

Then $(S,\mathscr{C}(S))$ (or, shorter, S) is called ***wavelet subtree*** *of the wavelet tree T.*

These wavelet subtrees are also wavelet trees.

The information contained in the coefficients is highly redundant. It suffices e.g. to know the coefficients at the root to be able to obtain the complete sets of coefficients in all nodes by wavelet decomposition; on the other hand, knowing the coefficients at all leaves is enough information as well, because all coefficients at all leaves can be obtained by wavelet reconstruction. More general, we have:

Theorem 5.1 (Sufficient independent sets of coefficients)**:** *Let $(T, \mathscr{C})$, $\mathscr{C} := (c_{u,z})_{u \in V, z \in \mathbb{Z}^d}$ be a wavelet tree. Let S be any subtree of T with the same root r as T. Then the coefficients associated to the leaves of S are sufficient to determine the coefficients at all nodes in T.*

Proof. Going upwards (direction root) by reconstruction, going downwards (direction leaves) by decomposition. □

Special cases:

- $S := \{r\}$: Original function is given, decomposition at leaves is obtained.
- $S := T$: Decomposed wavelet coefficients are given, reconstructed function is obtained.

Sparse coefficient spaces

The sparsity property of nonlinear wavelet approximation says that most coefficients after decomposition are zero. We want to use this property in our implementation and thus store only those coefficients which are non-zero. This procedure is similar to managing sparse matrices. The implementation is done by using key-value pairs. We refer to this data structure as to ***sparse coefficient spaces*** or short ***sparse spaces***.

The sparsity is also utilized for decomposition and reconstruction: we anyway cannot use the FWT algorithm (quadrature mirror filters) in its original form because neither the complete function will be known nor all coefficients will be used at once.

The procedure for reconstruction is as follows (needed in the identification procedure and for simulation with a given or identified model):

- Reconstruction is usually needed for values in some (discrete) hypercuboid (often only one value);
- Determine recursively which coefficients are needed;
- Begin with the coefficients at the leaves of the wavelet tree;
- Construct sparse spaces for each parents: Multiplicate each nonzero coefficient in the leaves with the corresponding filter coefficient and add it to the corresponding coefficient in the sparse space of the parents;
- Proceed in the same way going upwards in the tree;
- At the end a sparse space of the root is constructed: This gives the reconstructed function.

The procedure for decomposition is similar to reconstruction (usually not needed for the algorithm; we only need it for comparison in the examples we will present later):

- Procedure is similar to reconstruction, but beginning at the root and using the decomposition filters;
- At the end, the leaves contain the coefficients; they may not be very sparse, but many coefficients will be near zero;
- Apply an appropiate thresholding to restore sparseness.

Discrete approximation of the intermediate transition functions in a hypercuboid

Let f be any of the intermediate real transition functions $a_{k,j}$ or any of the primitive Preisach functions $F_{k,j}$ which need approximation (estimation) with wavelet coefficients. If f is a function from $\mathbb{R}^d$ to $\mathbb{R}^n$ for some dimensions d and n (in the state space, in the Preisach plane, etc.), we may look at the components f_i, $i = 1,\dots,n$, of f separately, and therefore assume that $n = 1$, i.e. $f : \mathbb{R}^d \longrightarrow \mathbb{R}$. Only a finite number of coefficients can be estimated. Therefore we have to restrict the estimation of f to a bounded hypercuboid $\tilde{Q}$ on $\mathbb{R}^d$. This is similar to e.g. the LOLIMOT algorithm which also works on a compact hypercuboid. To be able to use the Fast Wavelet Transform on f, we have to discretize $\tilde{Q}$. To do this, we choose a hypercuboid $Q \subseteq \mathbb{Z}^d$, a regular real $\mathbb{R}^d \times \mathbb{R}^d$-matrix as scaling matrix (usually diagonal), and define the rounding function $\rho(x) : \mathbb{R}^d \longrightarrow \mathbb{Z}^d$ componentwise by $\rho_j(x) := \lfloor x_j + 0.5 \rfloor$ for all $j = 1,\dots,d$ such that

$$Q = \rho(R\tilde{Q}).$$

We thus have defined a discrete grid $R^{-1}Q \subseteq \tilde{Q}$. The estimation task is then to find an appropriate map $h : Q \longrightarrow \mathbb{R}$ such

$$f(x) \approx h(\rho(Rx)) \qquad \text{for all } x \in \tilde{Q},$$

i.e. to find h such that

$$f(R^{-1}\xi) \approx h(\xi) \qquad \text{for all } \xi \in Q.$$

One should take care that during identification and simulation no extrapolation occurs, i.e. that no values fall outside the given hypercuboid. To avoid unexpected behaviour when it happens, values outside of the boundaries shall be projected to the nearest boundary. This gives a similar effect as the use of decision-tree based weight functions with the LOLIMOT algorithm.

Inner and outer hypercuboids in the wavelet tree Given a wavelet tree $(T, \mathscr{C})$, a discrete bounded hypercuboid $Q \subset \mathbb{Z}^d$, and a function $h : Q \longrightarrow \mathbb{R}$. We want to represent the function h through the coefficients given at the leaves of the wavelet tree T. To represent the function h exactly inside the hypercuboid Q, one has to compute and to store certain coefficients at the leaves of the wavelet tree. If we set $Q_r := Q$ for the root r, we can define

hypercuboids Q_u for each node $u \in V$ in the tree through recursive decomposition, such that the Q_u have minimal size but still ensure correct reconstruction of the coefficients contained in Q. In subsection 4.1.5 in the paragraph "Computation of number of coefficients", we showed how to compute the boundaries of these hypercuboids in each dimension for tensor product wavelets. The hypercuboids shrink in size if we go in direction of the leaves. There is a minimal size for the hypercuboids, only depending on the length of the decomposition filter. For tensor product wavelets, the minimal length of one edge of the hypercuboids is given by the length of the one-dimensional filter minus 1. If the height of the tree is sufficient, we get this minimally sized hypercuboids at the leaves.

If we begin with the hypercuboids Q_u at the leaves of the tree, we could go in the opposite direction and construct hypercuboids $\tilde{Q}_u$ for all nodes $u \in V$ via reconstruction, such that $\tilde{Q}_u$ is minimal in the sense that decomposition remains correct for the coefficients in the hypercuboids Q_u at the leaves. The boundaries are then increasing if we go in direction root, and we have $Q_u \subseteq \tilde{Q}_u$ for all $u \in V$. For tensor product wavelets where the univariate filters have length greater than 2, this inclusion is strict. We will call the Q_u ***inner hypercuboids*** and the $\tilde{Q}_u$ ***outer hypercuboids***. The outer hypercuboids $\tilde{Q}_u$ are needed during the reconstruction process, to ensure an exact reconstruction of the coefficients on the boundaries of the inner hypercuboids.

Example: Let be given at the root of a wavelet tree a hypercuboid of one dimension (interval); the left boundary (minimal position) shall be at 0, the right boundary (maximal position) at 63. We have thus 64 coefficients; if we had several dimensions and use tensor product wavelets, the following numbers would apply to each dimension separately. In table 5.3 we show for different filter lengths the boundaries of the hypercuboids at the nodes through different levels. In the upper rows (until the middle part), we see the inner boundaries obtained in each level during decomposition until the minimal number of coefficients is reached (at level 6); in the lower rows, the outer boundary for the additionally needed coefficients during decomposition are shown. These additional coefficients are needed during the decomposition process but can be neglected afterwards. From the table, it is easily recognizable that from some coarse level on, the minimal boundary of the inner hypercuboids stagnizes at a value of 2 minus filter length. From this, it is seen that the original hypercuboid could be extended such that the minimal boundary is set to this value (instead of 0), without the need to increase the number of levels. The computations for these additional points remain correct. In contrast, the upper boundary is already maximal and cannot be increased without the need of additional levels.

The computation of wavelet trees is only correct inside a predefined hypercuboid. One has to pay attention to points falling outside the hypercuboid! The behaviour of the wavelet trees is not predictable in these regions. Therefore we project the points to the nearest boundaries (axis-parallel).

Decision rule: Mean of wavelet trees

SMC methods lead to a discrete approximation of the distribution of the wavelet coefficients. We actually obtain distribution of wavelet trees. How to obtain concrete values for the coefficients, i.e. point estimates? One possibility is to use the ***mean of wavelet trees***:

		filter length = 2		filter length = 4		filter length = 6		filter length = 8	
	Level	min	max	min	max	min	max	min	max
Inner Boundaries	0	0	63	0	63	0	63	0	63
	1	0	31	-1	31	-2	31	-3	31
	2	0	15	-2	15	-3	15	-5	15
	3	0	7	-2	7	-4	7	-6	7
	4	0	3	-2	3	-4	3	-6	3
	5	0	1	-2	1	-4	1	-6	1
	6	0	0	-2	0	-4	0	-6	0
Outer Boundaries	5	0	1	-4	3	-8	5	-12	7
	4	0	3	-8	9	-16	15	-24	21
	3	0	7	-16	21	-32	35	-48	49
	2	0	15	-32	45	-64	75	-96	105
	1	0	31	-64	93	-128	155	-192	217
	0	0	63	-128	189	-256	315	-384	441

Table 5.3: Inner boundaries (upper rows) and outer boundaries of hypercuboids (lower rows) for partial reconstruction

- If all wavelet trees belonging to the particles have the same structure: Just take the mean of all coefficients corresponding to the same node and (key) index.
- Else: More difficult; trees could be brought to the same structure by decomposition and/or reconstruction (expanding or collapsing nodes).

Since the wavelet transform is linear, taking the mean of the wavelet coefficients at the leaves and reconstruct the function is the same as first reconstruct the functions and then taking the mean of the function values.

Problems with wavelet coefficients

There are two major problems which are actually due to the introduction of artificial dynamics (jitter):

- The locality of wavelet coefficients in connection with the recursive identification scheme leads to “uncontrolled” behaviour of those wavelet coefficients which are not involved in the identification step. This is the case if at a given time the output does not depend on the value of the considered coefficient. A possible remedy is to use the artificial dynamics only with coefficients which have an effect on the output.
- The drifting of wavelet coefficients: if wavelet coefficients of two different wavelet trees are “connected in series” and highly negatively correlated, the values in both trees may grow towards $-\infty$ and $+\infty$ respectively. This is a kind of non-identifiability. As a remedy the jitter may be equipped with a probability distribution which is biased towards zero. But it seems better to avoid such situations completely.

Guidelines for identification with wavelet trees

The following guidelines should be followed when wavelet trees are used for identification or simulation purposes:

- Choose an appropriate order (and thus smoothness) of the wavelet; Daubechies wavelets with filter length 2 (e.g. Haar wavelets) are usually not sufficient; linear functions can be approximated sparsely with Daubechies wavelets of filter length 4.
- Choose an appropriate granularity of the grid $R^{-1}Q$, i.e. choose an appropriate scaling matrix R and an appropriate size for the bounded hypercuboid $Q \subseteq \mathbb{Z}^d$; a coarse discretization increases the variances which leads to unstable identifications; the finer the discretization the smaller is the additional variance.
- Choose appropriate boundaries for the hypercuboids $\tilde{Q}$ (and thus for each Q); outside the boundaries the coefficients cannot be controlled by identification; therefore no evaluations should be done outside the boundaries.
- Choose an appropriate height of the wavelet tree; for sparsity reasons, the tree should be as heigh as possible; there is a maximal height which is around $\log_2$ of the grid size (in one dimension and if separable wavelets are used with dilation factor 2); but the heigher the tree, the more computations are needed for reconstruction.

The last three points, granularity, boundaries, and height of tree, are intimately connected: granularity and boundaries determine the grid size, whilst the height of the tree should be choosen according to the grid size. It should be remarked that increasing the grid size and increasing the tree height simultaneously such that the height is maximal does not change the number of non-zero coefficients to be estimated, and thus does increase the computational burden only marginally.

5.2.3 Implementation

We implemented a prototype of the described algorithm under the working name ***HySyWaT***, which stands for **Hy**steretic **Sy**stems with **Wa**velet **T**rees, in the programming language Java. The complete implementation has been done from the scratch, because the basic algorithms SISR and wavelet filters had to be modified in such a way that no existing implementations could have been used.

5.3 Examples

Before we present the results of the identification of a real shock absorber, we will consider three examples which shall bring an understanding of the behaviour of our identification procedure. Each example will serve to some different aspect on advantages and disadvantages.

We should note that the estimation of wavelet coefficients in the finer levels showed to be difficult due to the independence prior of the wavelet coefficients and their locality: Whereas in

the coarser levels the coefficients determine the functions more globally, non-zero coefficients in finer levels appear as located disturbances of the function, especially in higher dimensions. Imagine a digital picture decomposed with wavelets; areas of similar shading lead to areas of similar shading also in the decomposed scales; the coefficients are thus not independent from their neighbours. To avoid this problem, we will restrict our estimations to the coarsest levels. A possible solution of this problem may be given in the future by a spatial decomposition of the coefficient hypercuboids with decision trees as it is done in the LOLIMOT algorithm, or by introducing spatial dependencies in a stochastic way. Nevertheless, we could see this restriction to the coarsest level as a kind of introducing prior knowledge: We know that our functions do not have any singularities, so the coefficients of this level should be enough for estimating the rather smooth functions assumed here.

In all examples we used the SISR algorithm with state transition density as proposal. A resampling step is done after the Effective Sampling Size estimator (see subsection 3.6.4) is below a threshold which is always taken to be $1/10$th of the number of particles.

In all examples, we kept the number of particles, the size of the trees and the hypercuboids, as well as the number of time steps relatively small, to avoid long computation times. All estimations need thus only a few minutes for running.

5.3.1 First example: Linear mass-spring-damper system

Consider the second-order system:

$$m\frac{d^2\xi}{dt^2} + R\frac{d\xi}{dt} + k\xi = u$$

with $\xi : \mathbb{R} \longrightarrow \mathbb{R}$, input force $u : \mathbb{R} \longrightarrow \mathbb{R}$, and constants $m, R, k \in \mathbb{R}$, where we want to observe $\xi(t)$ at the discrete time points $t = 1, 2, 3, \ldots$ with additive Gaussian measurement noise. We use a simple Euler discretization with forward differences:

$$\frac{d\xi(t)}{dt} \approx \frac{\Delta\xi(t)}{\Delta t} = \frac{\xi(t+\Delta t) - \xi(t)}{\Delta t}$$

where we choose the time step to be $\Delta t := 1$. This leads to

$$m\big[\xi(t+2) - 2\xi(t+1) + \xi(t)\big] + R\big[\xi(t+1) - \xi(t)\big] + k\xi(t) = u(t)$$

which is equivalent to

$$\xi(t+2) = \left(2 - \frac{R}{m}\right)\xi(t+1) + \left(\frac{R-k}{m} - 1\right)\xi(t) + u(t).$$

Setting $x(t) := \begin{pmatrix} \xi(t+1) \\ \xi(t) \end{pmatrix}$ and adding some Gaussian process noise leads to

$$x(t+1) = Ax(t) + \begin{pmatrix} u(t) \\ 0 \end{pmatrix} + V(t), \qquad y(t) = Cx(t) + W(t)$$

with $V \sim \mathcal{N}_2(0, \Sigma_V)$, $W \sim \mathcal{N}(0, \sigma_W^2)$, where we choose

$$\Sigma_V := \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix}, \qquad \sigma_W^2 := 0.1,$$

and set

$$A := \begin{pmatrix} 2 - \frac{R}{m} & \frac{R-k}{m} - 1 \\ 1 & 0 \end{pmatrix}, \qquad C := \begin{pmatrix} 1 & 0 \end{pmatrix}.$$

Especially we choose:

$$m := 1, \qquad k := 1, \qquad R := 1.1, \qquad \text{hence} \qquad A := \begin{pmatrix} 0.9 & -0.9 \\ 1 & 0 \end{pmatrix}.$$

As excitation signal u, we will apply a random white noise. The initial distribution is given as $\mathcal{N}_2(\mu_0, \Sigma_0)$ with

$$\mu_0 := \begin{pmatrix} 4 \\ 2 \end{pmatrix} \qquad \text{and} \qquad \Sigma_0 := \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.4 \end{pmatrix}.$$

Even though the states could be estimated with the Kalman filter if the parameters are known, the joint estimation of parameters and states is not possible with the Kalman filter. Thus, even in this simple linear Gaussian model, approximation methods are necessary (nevertheless, Rao-Blackwellisation could be used, see subsection 3.6.4).

We want to estimate the upper row of A, i.e. the values $a_{11} := 2 - \frac{R}{m}$ and $a_{12} := \frac{R-k}{m} - 1$; all other values are assumed to be known. We produced observations by simulating the system with complete parameter set, such that the "true" values to be estimated are $a_{11} = 0.9$ and $a_{12} = -0.9$ respectively.

We use two estimation methods:

- Estimation of the real parameters a_{11} and a_{12} with SMC methods but without wavelet trees, and
- estimation of the function $x_1(t+1) = A_1(x_1(t), x_2(t))$ with a wavelet tree, where $x(t) = (x_1(t), x_2(t))^\top$; note that the "true" function is given by $A_1(x(t)) = a_{11}x_1 + a_{12}x_2$.

Remark:

- The wavelet tree used here has tow inputs: $x_1(t)$ and $x_2(t)$; thus the function A_1 and the estimated coefficients can be represented graphically.
- If we wanted to include u in a non-additive way we could use a wavelet tree with three inputs: $u(t)$, $x_1(t)$, and $x_2(t)$.
- If we wanted to estimate also the second row of A, we would need a second wavelet tree representing the function $x_2(t+1) = A_2(x_1(t), x_2(t))$.

In the estimation with wavelet tree, the estimated function is nonlinear. To avoid the estimation of a function which is too nonlinear, we use the Daubechies wavelets with filter lengths 4. Thus, the first two moments vanish, and linear functions on a compact intervall can be represented exactly with finitely many coefficients. We also use a usual wavelet tree with maximal height, such that at the four leaves of the deepest level, we have hypercuboids with minimal edge lengths 3. We further restrict the estimation to the coefficients which belong to the leaf with the scaling coefficients; this is the leaf at the deepest level with the enumeration 0. We thus have to estimate 9 coefficients. Compare this to the 2 parameters in the estimation of the linear function without wavelet tree.

Results For the estimation of the parameters, we used 20 000 particles and only 64 time steps. We added also a small jitter to the parameters. In figure 5.1, we show the histograms of the empirical distribution of the estimated parameters in the last time step. In figure 5.2, we plot the means of the empirical distribution over time. We see that already after a few time steps the estimated parameters are quite near the original ones; due to the jitter, they are time-variant, but nevertheless, fluctuations are small. In figure 5.3, the states and observations of the original simulation are compared to a simulation with the estimated parameters. As parameter estimate, the mean of the empirical distributions at the last time step was taken. The fit is quite good. Other runs generally showed similar results. Nevertheless, in some runs the system exploded, due to an estimation of the parameters beyond the stability boundary. To prevent this, an application of Durbin-Levinson recursion could help (see subsection 3.4.2).

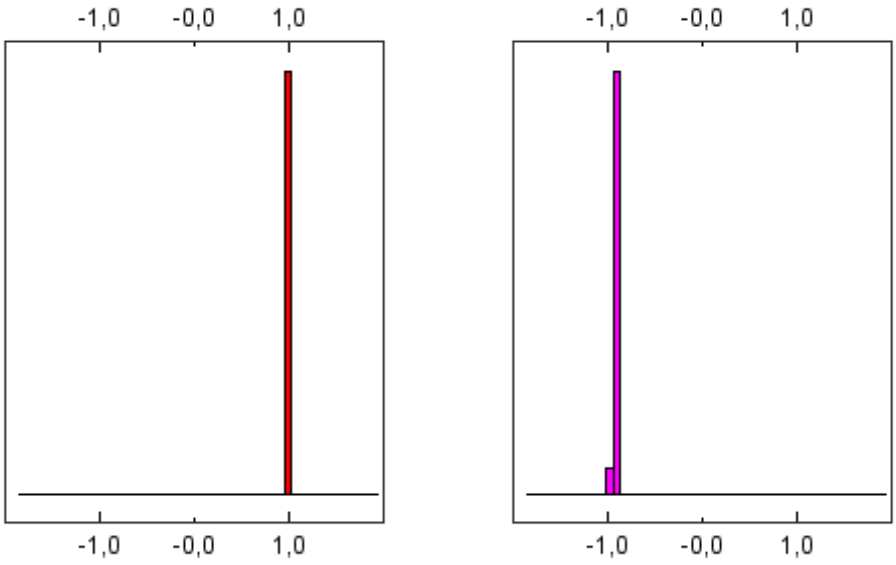


Figure 5.1: First example, estimation of scalar parameters: Histograms of estimated parameters

In figure 5.4, we show coulor plots of the coefficients of the two wavelet trees representing the original functions $A_1 = 0.9x_1 - 0.9x_2$ and $A_2 = 1x_1 + 0x_2$, respectively. In each plot, at the

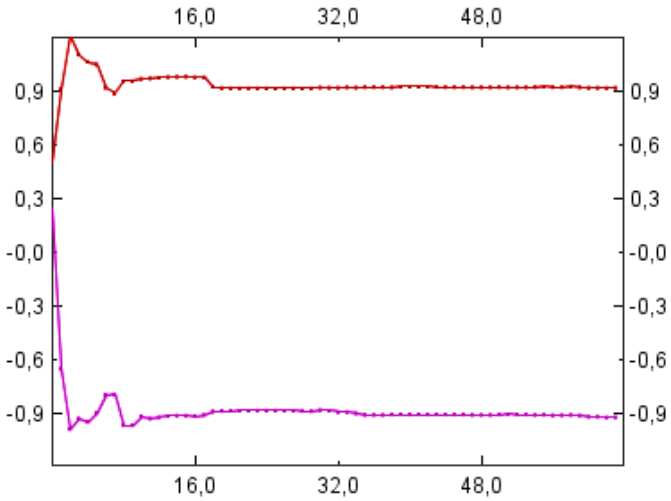


Figure 5.2: First example, estimation of scalar parameters: Plot of estimated parameters over time during estimation

top, the tree T is pictured; at the lower left, the decomposed inner hypercuboids at all leaves are depicted. Shown are their boundaries and the non-zero coefficients. The hypercuboids are arranged in such a manner as it is usual for image analysis with wavelets. The hypercuboids do not fit to each other because a filter length greater 2 is used; thus, there are empty spaces necessary between them in the picture. As can be seen, non-zero coefficients appear only in the 0-node at the coarsest level. This is due to the polynomial reconstruction property of the Daubechies wavelets with filter length 4: linear polynomials can locally be reproduced exactly with finitely many coefficients; these appear only as scaling coefficients in the coarsest level. The colours have been choosen such that negative values are blue, positive values are red, and light green is zero.

At the lower right of each plot, the reconstructed outer hypercuboid at the root is depicted. The corresponding inner hypercuboid is marked with a red frame. Also here, only the non-zero coefficients are coulored. We see that only the values in the inner hypercuboid represent correct linear functions. There are non-zero coefficients in the outer hypercuboid outside the inner cuboid. These are pictured just for demonstration purposes and will never be used for the simulations.

In figure 5.5 we zoom into the scaling coefficients (left), and show the corresponding inner hypercuboids of the root (right).

In figure 5.6, we depict the estimated wavelet tree. The estimated wavelet tree is given by the mean of the empirical distribution of the wavelet trees, i.e. the means of each coefficient, at the last time step. We again want to stress that only the inner hypercuboid contains the relevant coefficients. We depict these in comparison with the original ones in figure 5.7. In spite of the nonlinear estimation method, the estimation quality in this part is remarkable.

Eventually, in figure 5.8, we compare states and observations simulated with the original wavelet tree and with the estimated one.

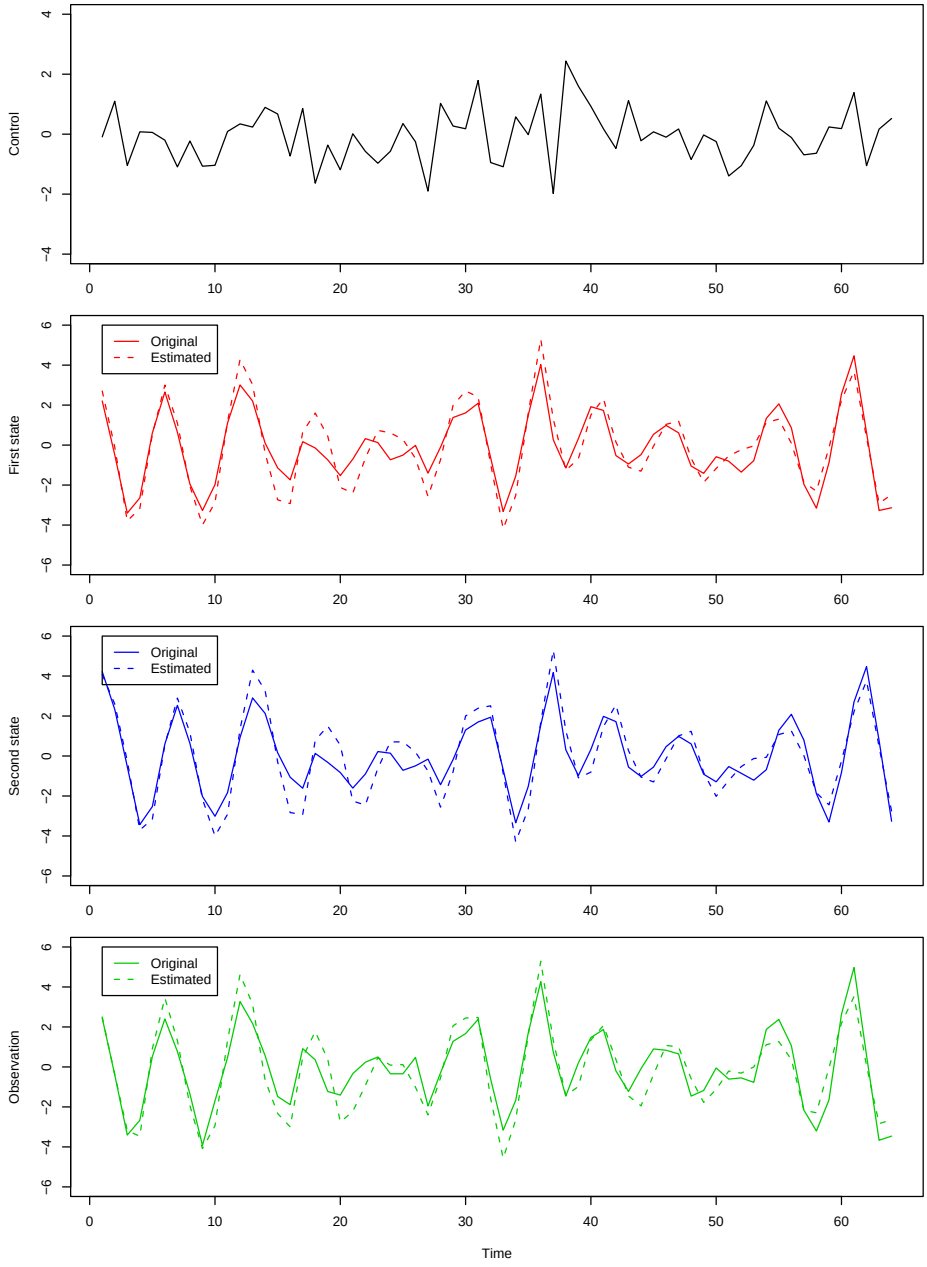


Figure 5.3: First example, estimation of scalar parameters: From top to bottom: input signal (black), two state signals (red and blue), and output signal (green); the solid lines depict simulations made with the original parameters, the dashed curves depict simulations made with the estimated parameters

For the wavelet estimation, we used only 2000 particles. Despite the low number of particles and the low number of time steps, the estimation is astonishingly good.

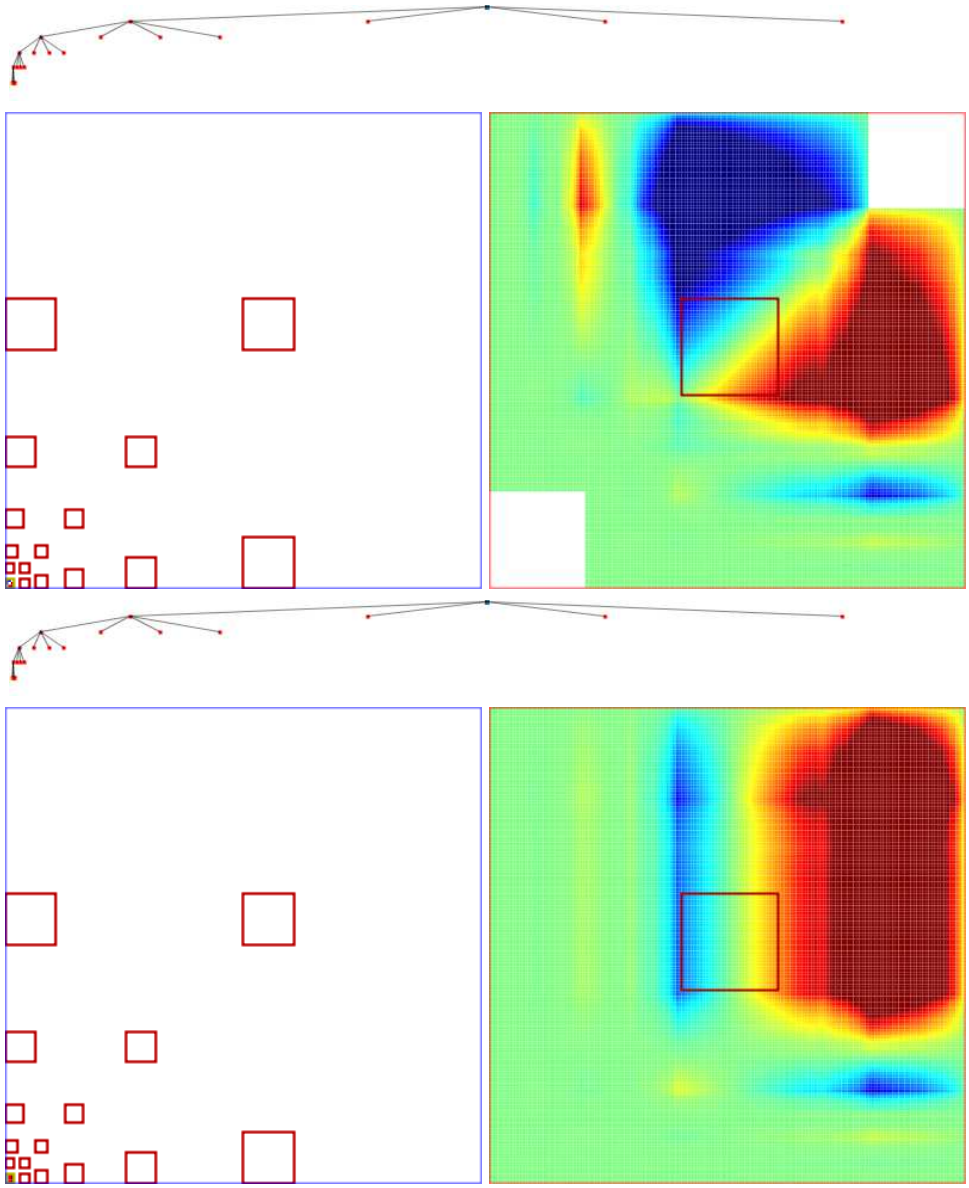


Figure 5.4: First example, estimation with wavelet trees: Decomposed and reconstructed coefficients of the wavelet trees realizing the original 2-dimensional linear function

5.3.2 Second example: Nonlinear mass-spring-damper system

We replace the constant k by a function depending on ξ. Thus, the linear function $k\xi$ is replaced by a nonlinear function $\kappa(\xi)$. We will use a κ which shows saturation effects for

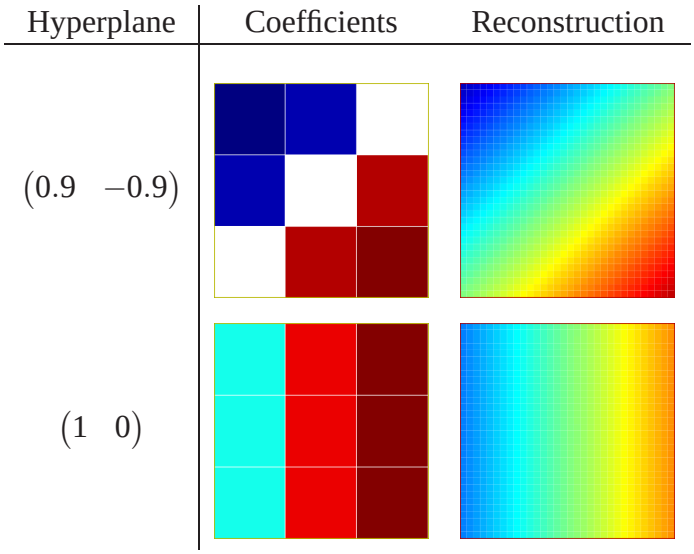


Figure 5.5: First example, estimation with wavelet trees: Zoom into reconstruction in inner hypercuboids (left) and coefficients of scaling leaf (right) for the original function

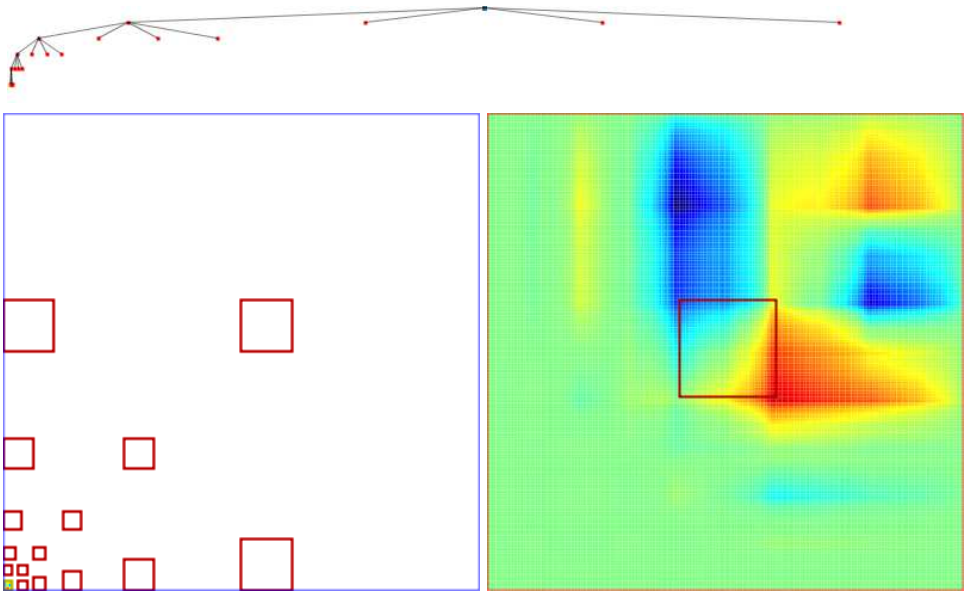


Figure 5.6: First example, estimation with wavelet tree: Decomposed and reconstructed coefficients of wavelet tree realizing the estimated 2-dimensional function

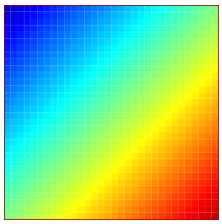
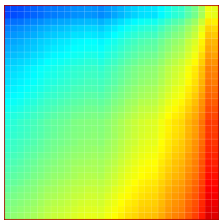
Hyperplane	Original	Estimation
$\begin{pmatrix} 0.9 & -0.9 \end{pmatrix}$		

Figure 5.7: First example, estimation with wavelet tree: Zoom into reconstruction in inner hypercuboids: original (left) and estimate (right)

large values of ξ; in particular we take the sigmoid function

$$\kappa(\xi) := \frac{2}{1+\exp(-\xi)} - 1$$

which asymptotes -1 for $\xi \to -\infty$ and $+1$ for $\xi \to +\infty$. We thus replace the matrix A by a nonlinear function h:

$$x(t+1) = \begin{pmatrix} x_1(t+1) \\ x_2(t+1) \end{pmatrix} = \begin{pmatrix} \left(2 - \frac{R}{m}\right) x_1(t) + \left(\frac{R - \kappa(x_2(t))}{m} - 1\right) x_2(t) + u(t) \\ x_1(t) \end{pmatrix}.$$

Estimation of the upper row shall again be done using a wavelet tree.

Results Analogously to the previous example, we show in figure 5.9 the wavelet tree for the original function h, in figure 5.10 the wavelet tree for the estimated function, and in figure 5.11 the comparison of the original and estimated coefficients of the reconstructed inner hypercuboid. In figure 5.12 we depict states and observations simulated with the original and estimated wavelet trees. We used only 2000 particles for the estimation. The estimation in this case is more difficult than before. To get better results, one could increase the height of the tree (leads to lower discretization errors), enlarge the hypercuboid (during the estimation of this example, many evaluations at the boundaries of the hypercuboid or even outside occurred), increase the number of particles, or increase the number of time steps.

5.3.3 Third example: Preisach hysteresis

We now consider a Preisach hysteresis system given by the primitive functions F, F_- and F_+ which is a scaled version of the system described in example (2) in subsection 2.2.2 with

$$F_T(\alpha, \beta) = \frac{1}{2}(\alpha - \beta)^2,$$

i.e. a two-dimensional quadratic function. We added also a small noise to the observations. The Preisach plane is always two-dimensional, thus we need a wavelet tree with two inputs,

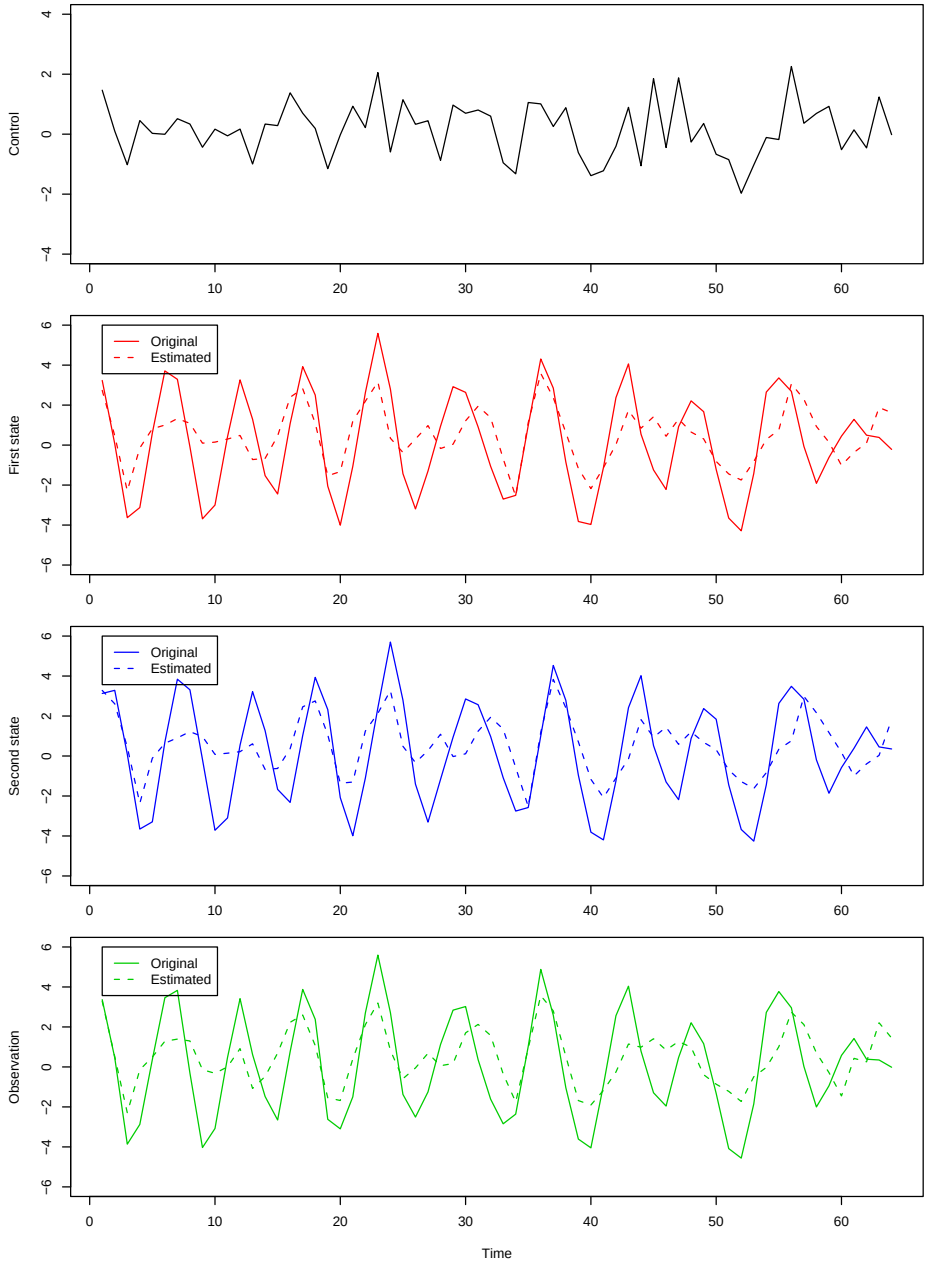


Figure 5.8: First example, estimation with wavelet tree: From top to bottom: input signal (black), two state signals (red and blue), and output signal (green); the solid lines depict simulations made with the original parameters, the dashed curves depict simulations made with the estimated parameters

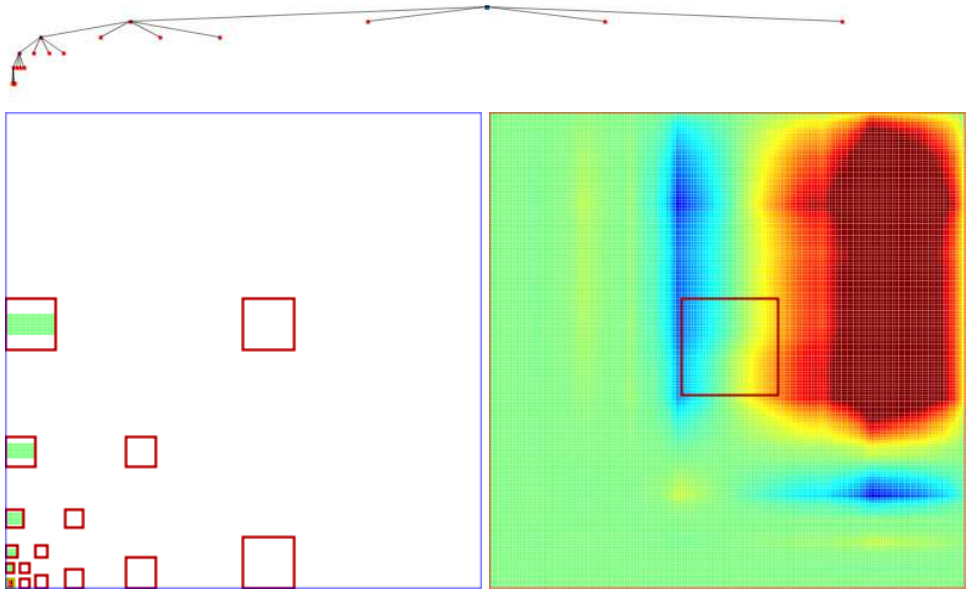


Figure 5.9: Second example: Decomposed and reconstructed coefficients of the wavelet tree realizing the original 2-dimensional nonlinear function

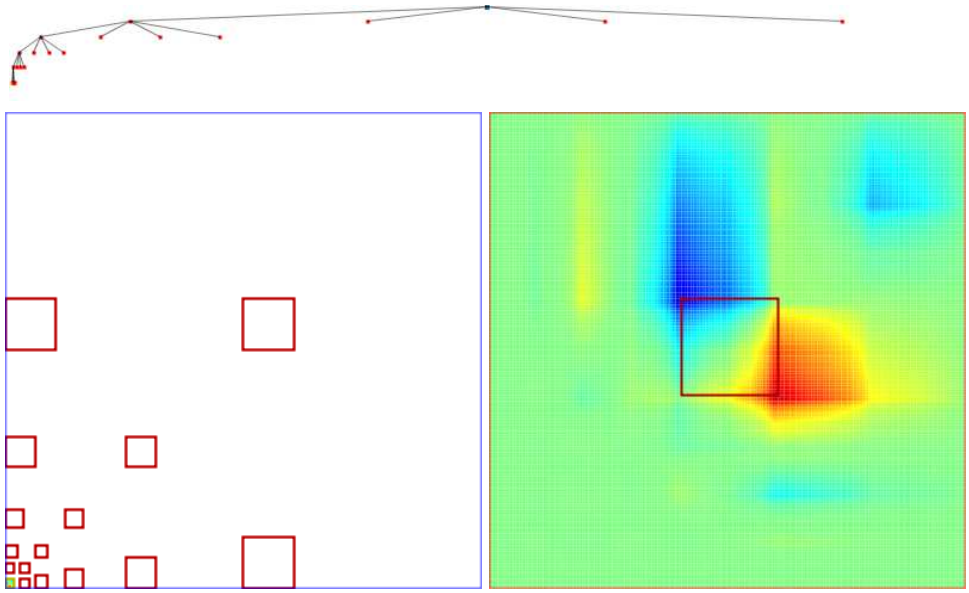


Figure 5.10: Second example: Decomposed and reconstructed coefficients of the wavelet tree realizing the estimated 2-dimensional function

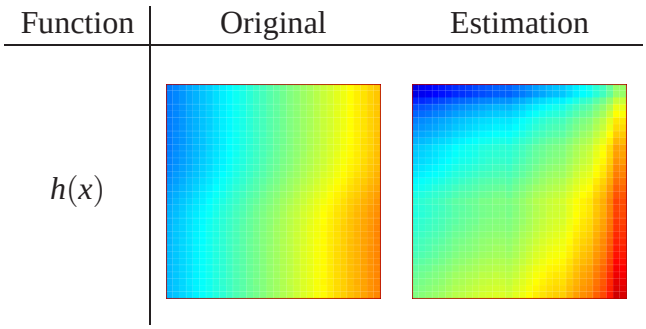


Figure 5.11: Second example: Zoom into reconstruction in inner boundaries: original (left) and estimate (right)

and we can represent this tree and its coefficients graphically. We additionally need two one-dimensional wavelet trees for F_+ and F_-. With Daubechies wavelets of filter lengths 4, it is actually difficult to represent quadratic functions, whereas with Daubechies wavelets of filter lengths 6, quadratic functions on a compact interval can be represented exactly with finitely many coefficients. Nevertheless, even Daubechies wavelets with filter length 4 used for the estimation give good results. We used only 200 particles in this case, and 400 time steps.

Results In figures 5.13 and 5.14, we show original and estimated wavelet tree for the function F, respectively (we do not show the estimations for F_- and F_+). In figure 5.15, we compare the inner hypercuboids, and in figure 5.16, we compare original and estimated observations. The estimation shows a quite good fit to the original simulations. Also the function on the Preisach half plane is very well estimated. One should note that it only needs to be estimated above the diagonal from bottom left to top right. The coefficients below the diagonal are arbitrary.

5.4 Identification of real data

5.4.1 The data

The data are from a real shock absorber. The measurements were taken by the LMS company and have been kindly provided for use in this thesis. The measurements consist of 4 data sequences taken from different experiments:

- one data sequence is measured during a road test,
- the remaining experiments have artificial inputs:
  - White noise,
  - Sine wave 6 Hz with changing amplitude,
  - Sine wave 0.1 Hz (quasi-static).

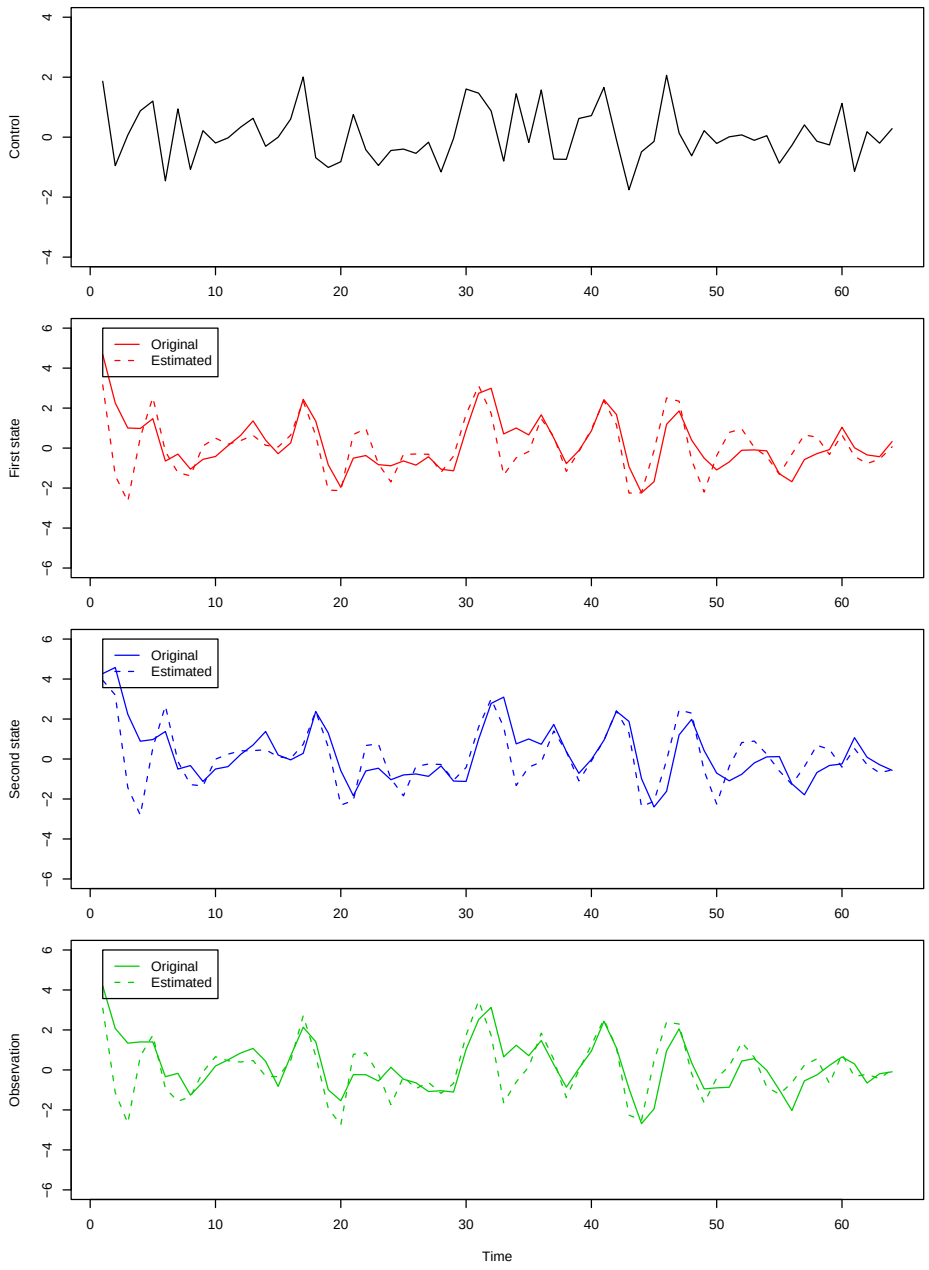


Figure 5.12: Second example: From top to bottom: input signal (black), two state signals (red and blue), and output signal (green); the solid lines depict simulations made with the original parameters, the dashed curves depict simulations made with the estimated parameters

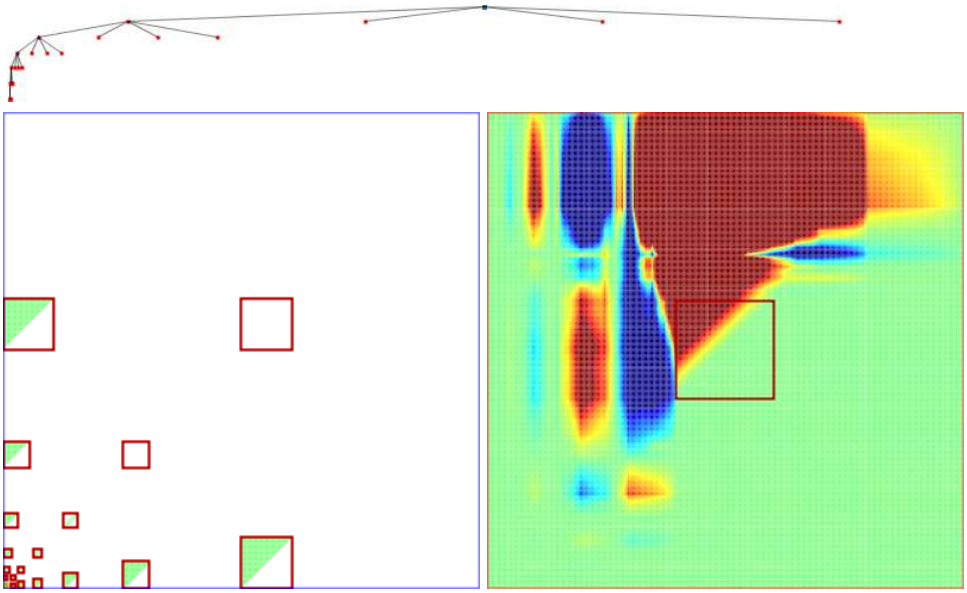


Figure 5.13: Third example: Decomposed and reconstructed coefficients of the wavelet trees realizing the original 2-dimensional primitive Preisach function

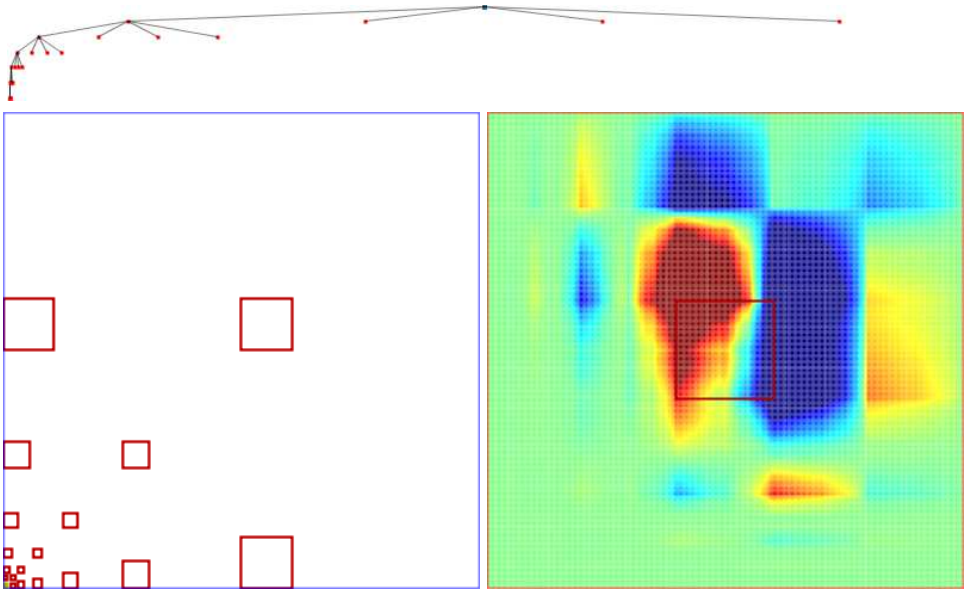


Figure 5.14: Third example: Decomposed and reconstructed coefficients of the wavelet trees realizing the estimated 2-dimensional primitive Preisach function

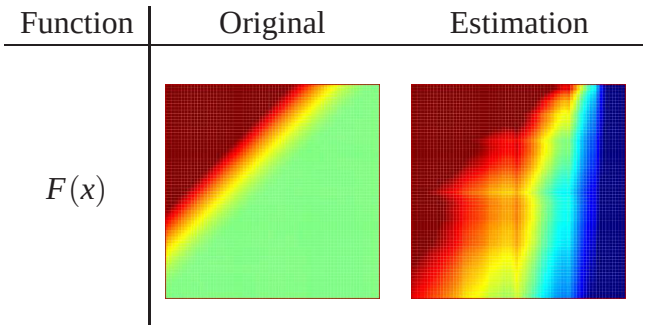


Figure 5.15: Third example: Zoom into reconstruction in inner boundaries: original (left) and estimate (right)

The data are pictured in figures 5.17, 5.18 and 5.19 in different representations. The data has been scaled, and some data sequences had a small offset, they have been slightly adjusted to 0, but otherwise the data remain unchanged.

Problems with the artificial input data are:

- The amplitude of white noise is very small compared to the amplitude of real (road test) input, but the amplitude could not have been increased without danger of destroying the test rig.
- The slow sine wave (0.1 Hz) is meant for identification of hysteresis; no inner loops occur, and the part of the state space which is covered by this signal is rather small: Since velocities are small, the covered area of the real part of the state space is more or less one-dimensional. The same is true for the Preisach plane. Similar statements hold for the 6 Hz signal.

Therefore, the road test signal is the essential signal for estimation: Being much more informative than the other sequences, it covers a much larger part of the state space. This can already be seen on the input/output diagram of figure 5.19. We nevertheless want to use all data sequences for the identification of the model.

Summarized, we have very heterogeneous data, and identification is difficult.

Simultaneous identification The idea used to circumvent the mentioned problems is to try simultaneous identification of all data sequences at once. This is easy with the SMC approach:

- Let run 4 systems in parallel: Each system is running on one of the input sequences.
- The outputs are independent given the parameters, we thus use the algorithm with joint parameters and individual states.
- Given parameters and states, the joint observation probabilities are equal to the product of the single individual observation probabilities because of conditional independence.

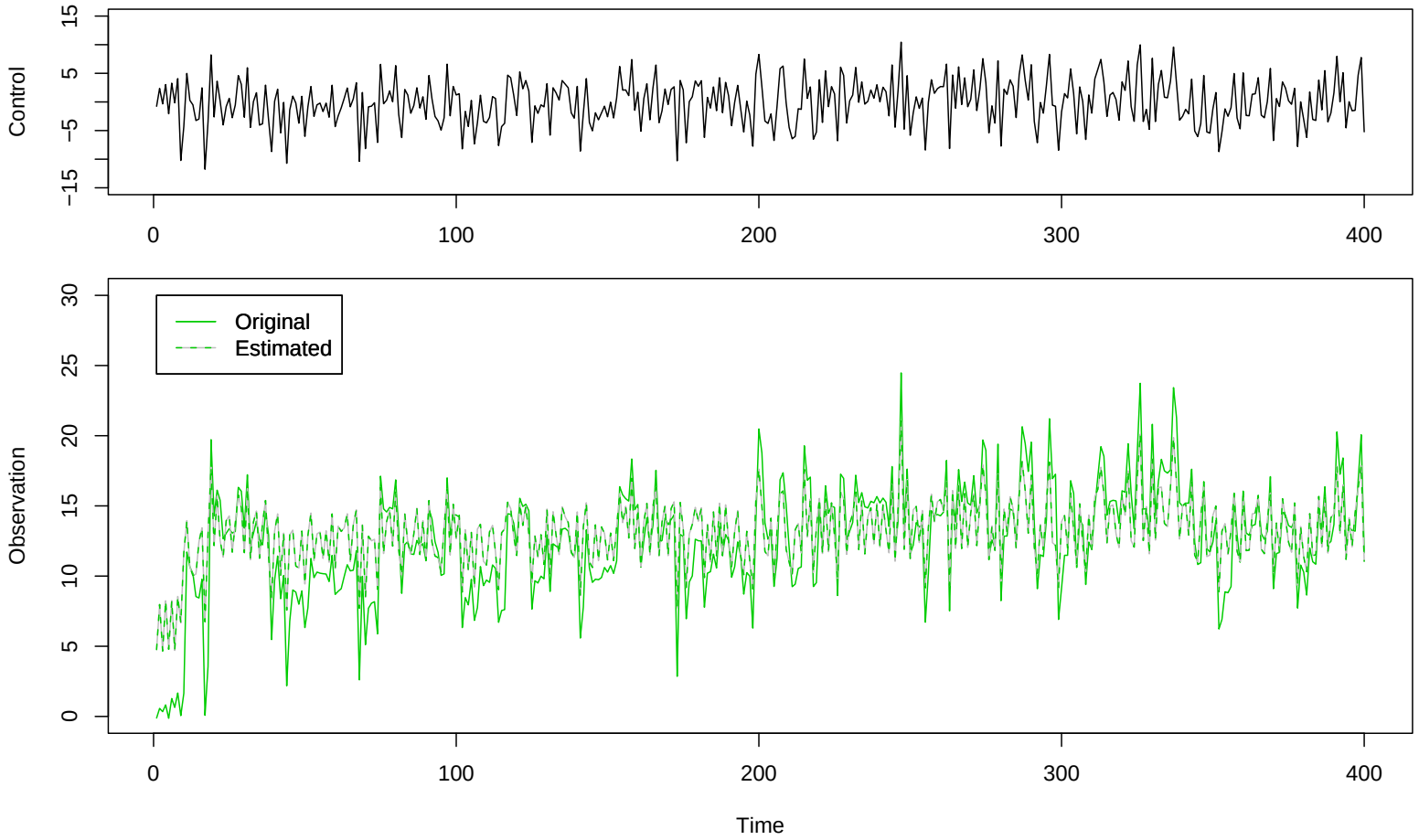


Figure 5.16: Third example: Input signal (black) and output signal (green) of the hyteresis; the output of the original system is depicted as solid line, the output of the estimated system is depicted as dashed line

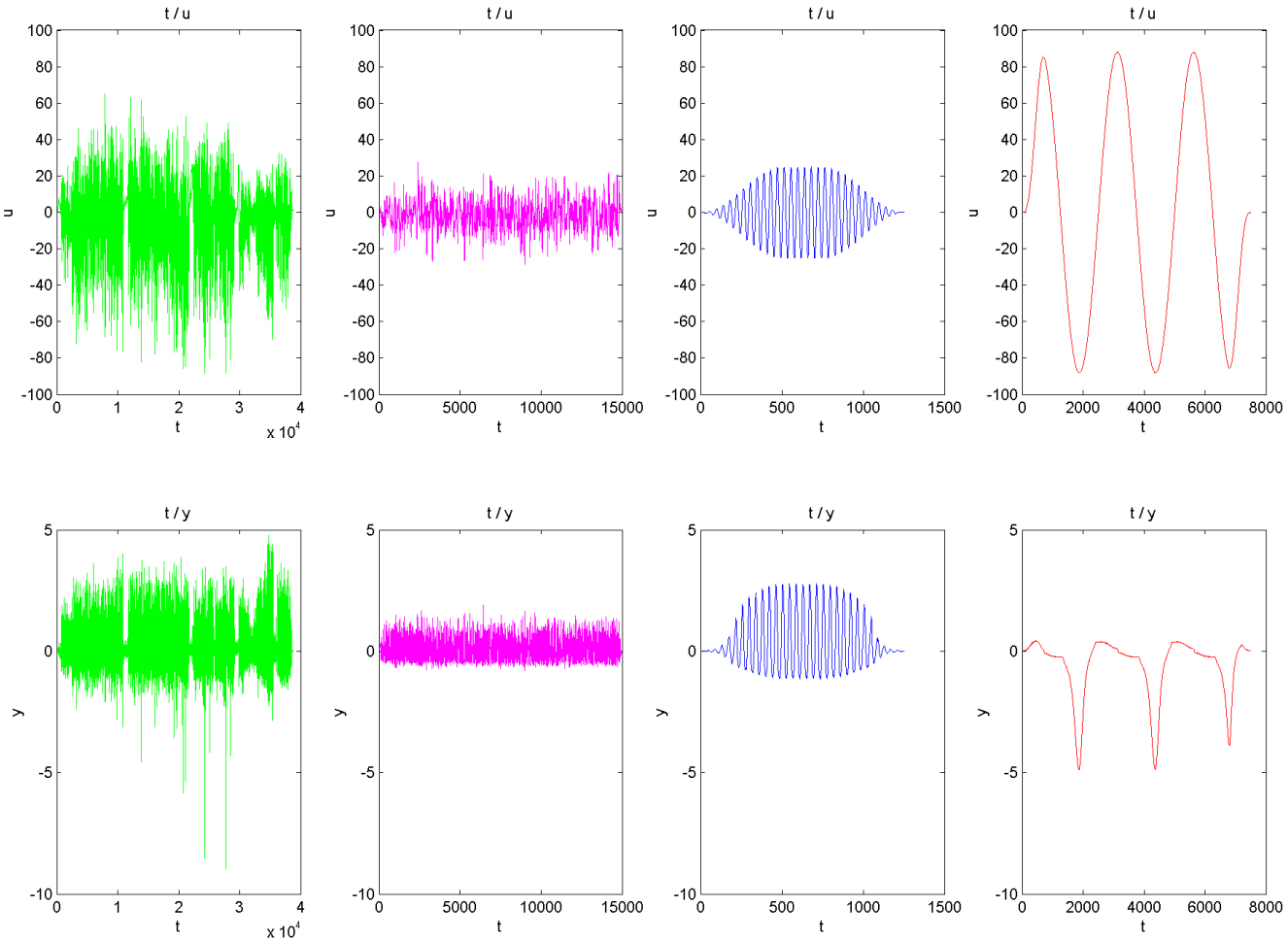


Figure 5.17: The measured data taken from experiments with a shock absorber, single data sets

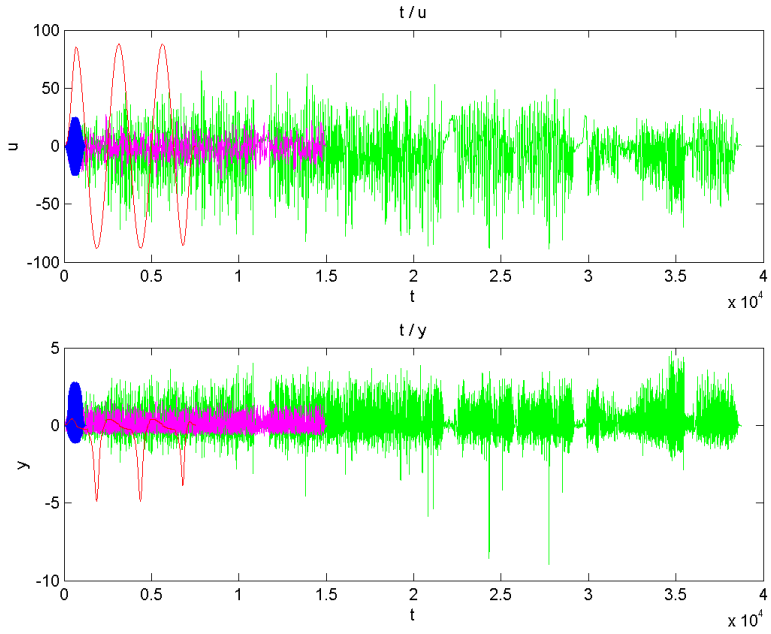


Figure 5.18: The measured data taken from experiments with a shock absorber, overlaid data sets

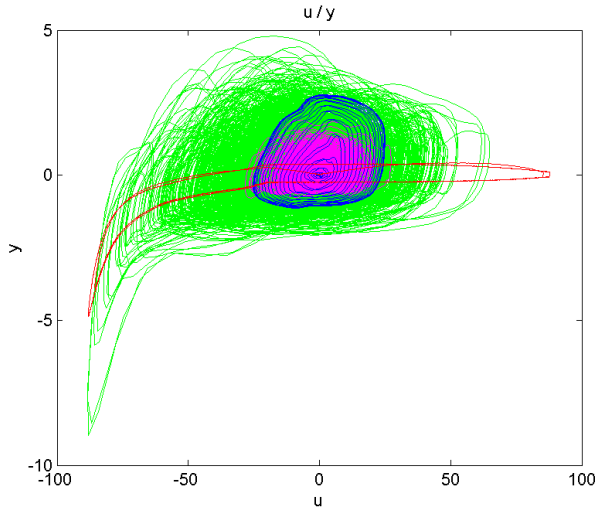


Figure 5.19: The measured data taken from experiments with a shock absorber, input/ouput diagram

Model The data showed extremely difficult to be estimated through a grey- or black-box model; it was impossible with the LOLIMOT algorithm, and the author tried several different models with the new algorithm. At last, the simplest of them was (at least to the most part) successful. It is a weighted parallel combination of a Preisach hysteretic system and a difference system, see figure 5.20; we call it the mixed model. In this figure, **P** denotes the Preisach system with prefixed alternating sequence r. The input is u, the output of it is the intermediate state $\tilde{x}_1$. In the lower part of the figure, we find a difference system with tapped delays, with input u and output $\tilde{x}_2$. **Tr** denotes a transform. Concerning this transform, we first used simply a multiplication with a constant factor C that had to be estimated, but we got better results by using a further one-dimensional wavelet tree where we estimated the coefficients. There are also two noise sources V and W. They were chosen as Gaussian noises with mean zero. The variance of the process noise was chosen to be very small, nearly neglectable. To be estimated are thus the three wavelet trees for the Preisach hysteresis, and the additional wavelet tree for the transformation. All wavelet trees have height 6 which corresponds to hypercuboids of size 64 (plus filter length minus 2) in each dimension. We used Daubechies wavelets with filter length 4. In this case, we estimated the coefficients of all leaves at the deepest level, i.e. the coarsest coefficients and the corresponding details. For each wavelet tree, we had thus $2 \cdot 3 = 6$ coefficients for the one-dimensional cases and $4 \cdot 9 = 36$ coefficients for the two-dimensional case, in summary $1 \cdot 36 + 3 \cdot 6 = 54$ coefficients to be estimated. To get smoother outputs for values near the diagonal of the Preisach plane, the outputs of the wavelet tree corresponding to the two-dimensional Preisach function $F(\alpha, \beta)$ have been multiplied by a sigmoid function on $|\beta - \alpha|$ which is zero at input zero and asymptotes 1 for large inputs. This smoothens the discretization error and avoids a jumping behaviour of the simulated output near the change points.

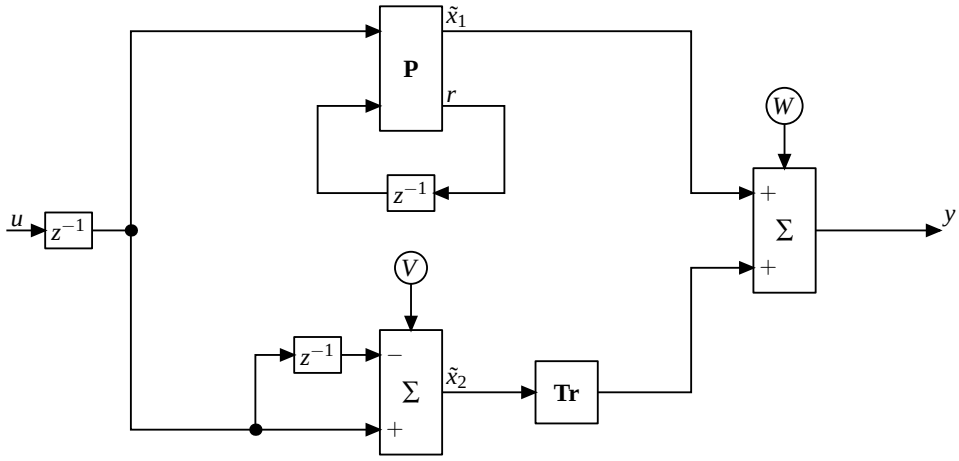


Figure 5.20: The mixed model with differential and hysteretical dynamics

Results We used only 20 particles to reduce computation time, and for identification, we used only the first 500 steps of the data sequences. To test the quality of the estimated model, we tried simulations also on data which where not used for identification. In figure 5.21, we

show the estimated inner hypercuboid of the reconstructed Preisach function. In figure 5.22, we depict the original and estimated outputs of the several data sequences for the 500 time steps used for identification. In figure 5.23, we try to give a rough impression on the quality of the simulations. We used the first 2500 steps of the data, the first 500 being the steps used for identification, the remaining observations being completely unknown to the system (remark that the third data sequence has only a length of 1251 steps, so we show only these steps). As a detailed example, we show in figure 5.24 a shorter sequence of 1200 steps of the road test data, i.e. the steps from times 24 900 to 26 100 after the start of the simulations, where in the measurements both rather large values as well as smooth sections occur. It should be remarked that the estimated outputs are pure simulations from step 0 until the shown time steps, without any adaptation to measured data! Taking into account the very small number of particles and the small number of time steps used for identification, and also the stochastic nature of the model, the simulated signal is astonishingly good for 3 of the 4 data sequences. Only the simulated output of the quasi-static 0.1 Hz signal is not well reproduced. The simulations are extremely stable. Considering the first three data sequences, the simulated output always follows at least roughly the measured output, and in many places we have a really close fit. Discrepances can be recognized especially if the output oscillates with a large amplitude, where the extreme values are not always reached or overshot. In the parts were the signals are flatter, a small offset can occur.

Several repetitions of the identification process showed that the results are always nearly identical. The identification is stable.

The bad fit for the fourth data sequence may have several reasons: First of all, one could argue that the part which is not fitted well was not used for identification. But also using the complete data sequence for identification does not help much, it even worsens the results for the remaining data sequences. The bad performance may be due to the sparse coverage of the Preisach plane, or an artefact induced by the artificial dynamics of the parameters/coefficients. If the complete data sequence is used for identification, it is clearly recognized that the estimated output flattens and reproduces a kind of averaged output over time of the original data sequence. This in turn may result from the time-varying estimation of the wavelet coefficients: they adapt too much to the observations at a given time and forget too much about the earlier observations (hence also the mentioned worsening of the other three sequences). Another possibility might have been that we identified only wavelet coefficients in the coarsest level of the wavelet tree which does not give enough flexibility for the reconstruction of the Preisach function. But several trials to use also coefficients at finer levels and with higher filter orders did not gain any improvement. The subject needs further investigation. The author expects that with further improvements of the algorithm this problem might be settled.

Summary of results

It seems that our new method is principally able to identify complex systems. Nevertheless, it has to be improved; the adjustments of jitter parameters, choice of grid size etc. are cumbersome and time-consuming. Also the quality of the estimations has to be improved. But the method should be seen as a first step only, and the partial estimation of the shock absorber data should be seen as a proof that the consideration of the method is not worthless. Especially the

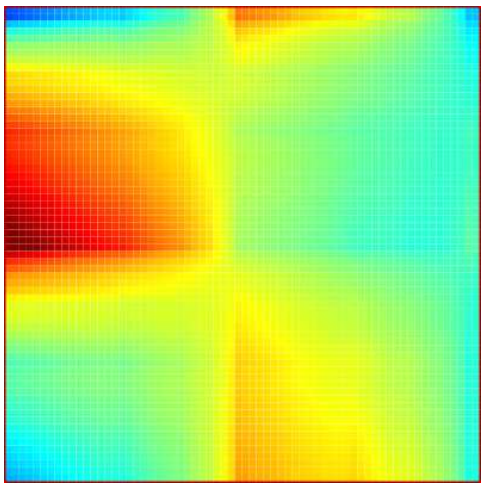


Figure 5.21: Data identification: Reconstructed inner hypercuboid of estimated Preisach function

road test data sequence of the shock absorber could already be simulated astonishingly well. And this is the sequence which really is important for applications. An important point is also that the simulations could be done in real time: only forward computations are necessary. Together with the obtained stability and accuracy (which surely can even be increased), the usage of the identified grey-box model as a part of a model for a whole axis or even a complete car may open the way to interesting applications.

5.5 Conclusion and future work

5.5.1 Résumé: Usage of identification methods

We try to explain, according to the distinct modelling approaches, when and which of the identification methods (improved LOLIMOT or newly introduced model class) is reasonably used.

Black-box approach

The original LOLIMOT algorithm must be seen as black-box approach, because there are actually no possibilities to include a-priori knowledge into the model. As mentioned earlier, this algorithm can be improved in many respects, without recourse to the new model class and the essentially more complex identification algorithm. We treat models with external dynamics (input/output models) and internal dynamics (state space models) separately.

External-dynamics models In the original algorithm, only global NARX models can be identified. The extension to other model structures like NOE models must be done by an additional gradient-based adaptation, applied during or after the original LOLIMOT run. In connection with the likewise supplementary pruning methods this procedure leads to smaller

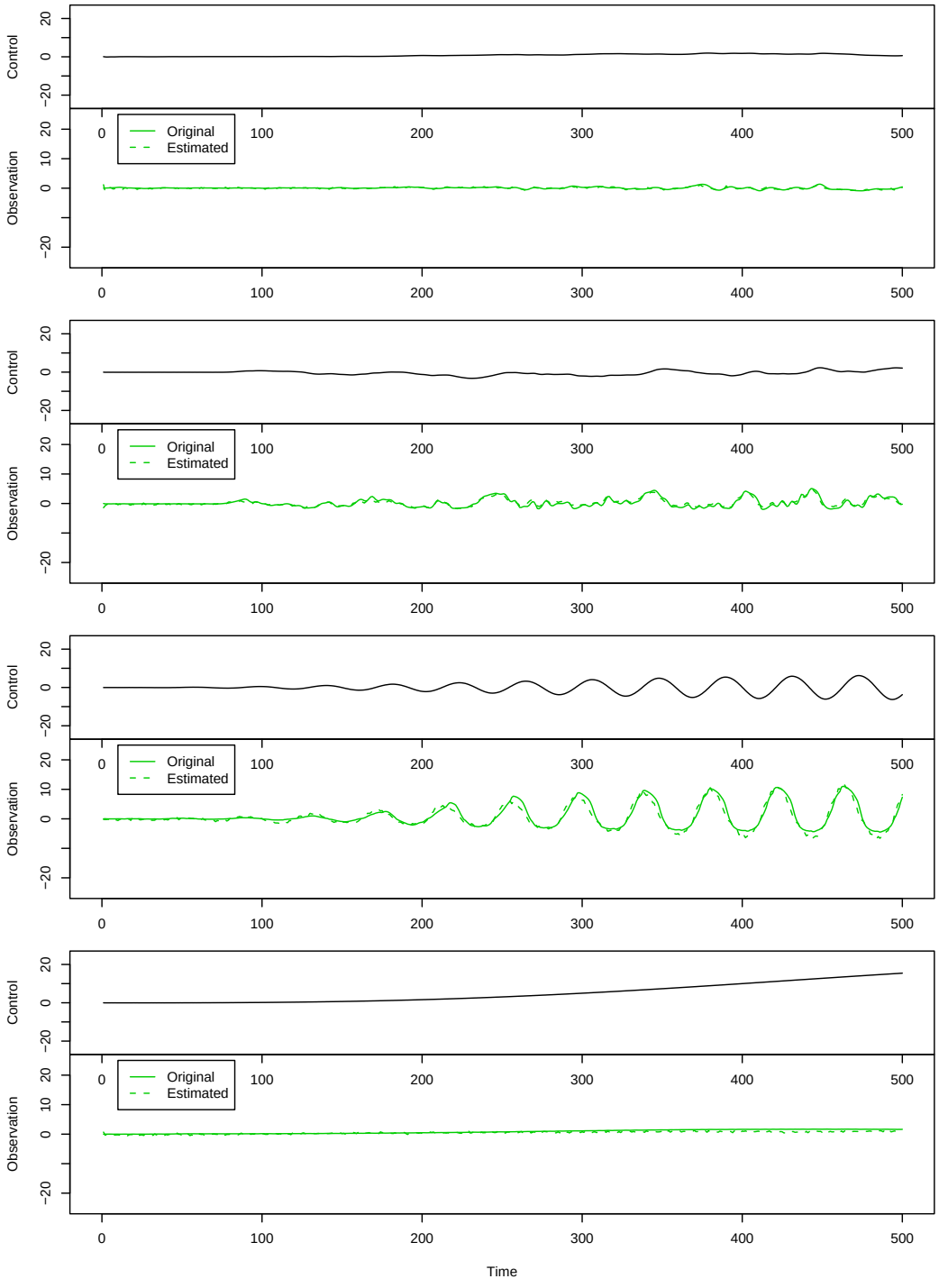


Figure 5.22: Measured and estimated outputs of data used for identification: solid green line: measured output; dashed green line: simulated output of the estimated model; black line: input signal; the signals consist of the first 500 time steps of each of the four data sequences

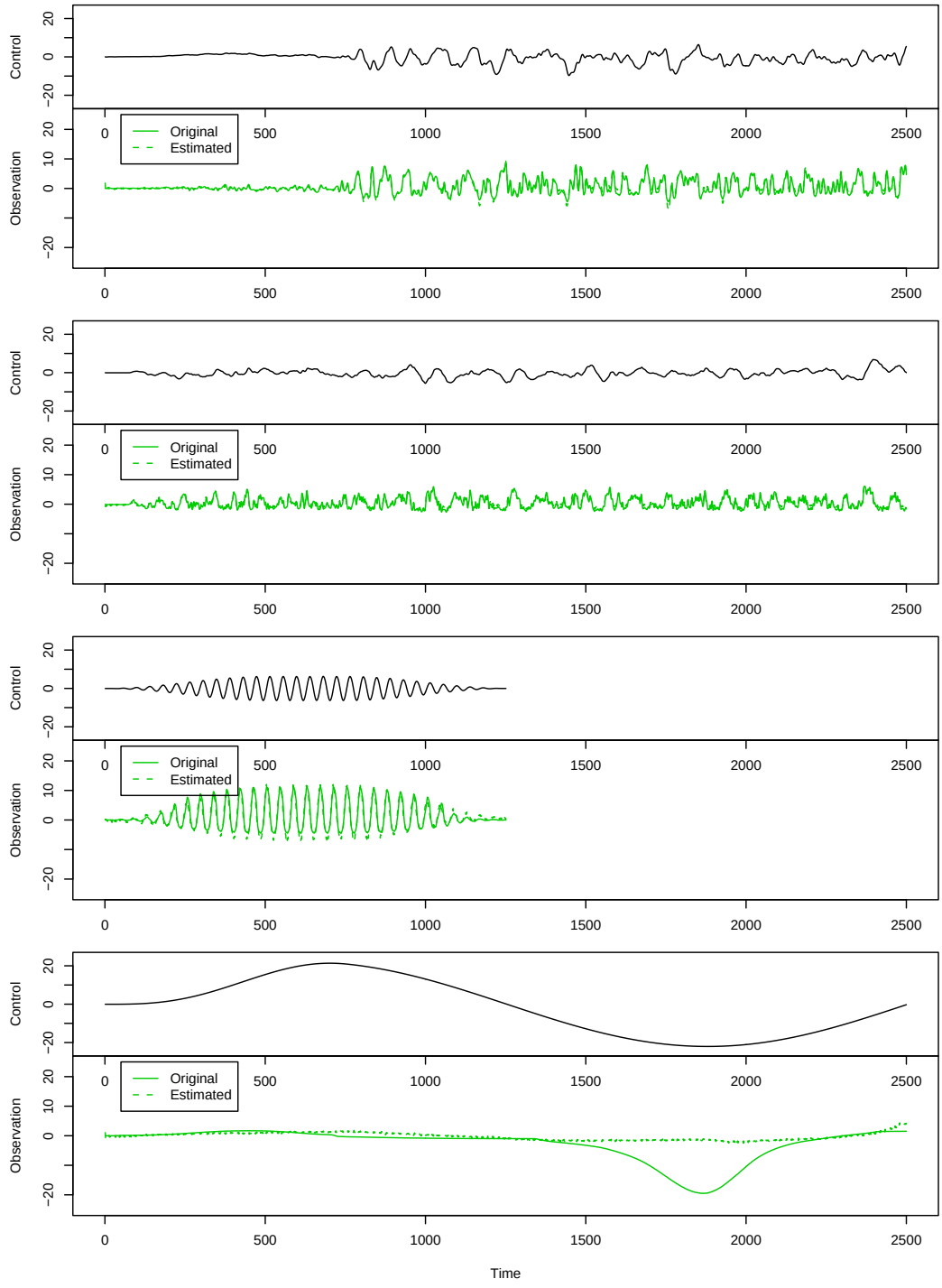


Figure 5.23: Measured and estimated outputs of data used for testing: solid green lines: measured output, dashed green lines: simulated output of the estimated model; black lines: input signal; shown are the first 2500 time steps of the first, second and last sequence, and the complete third sequence consisting of 1251 time steps

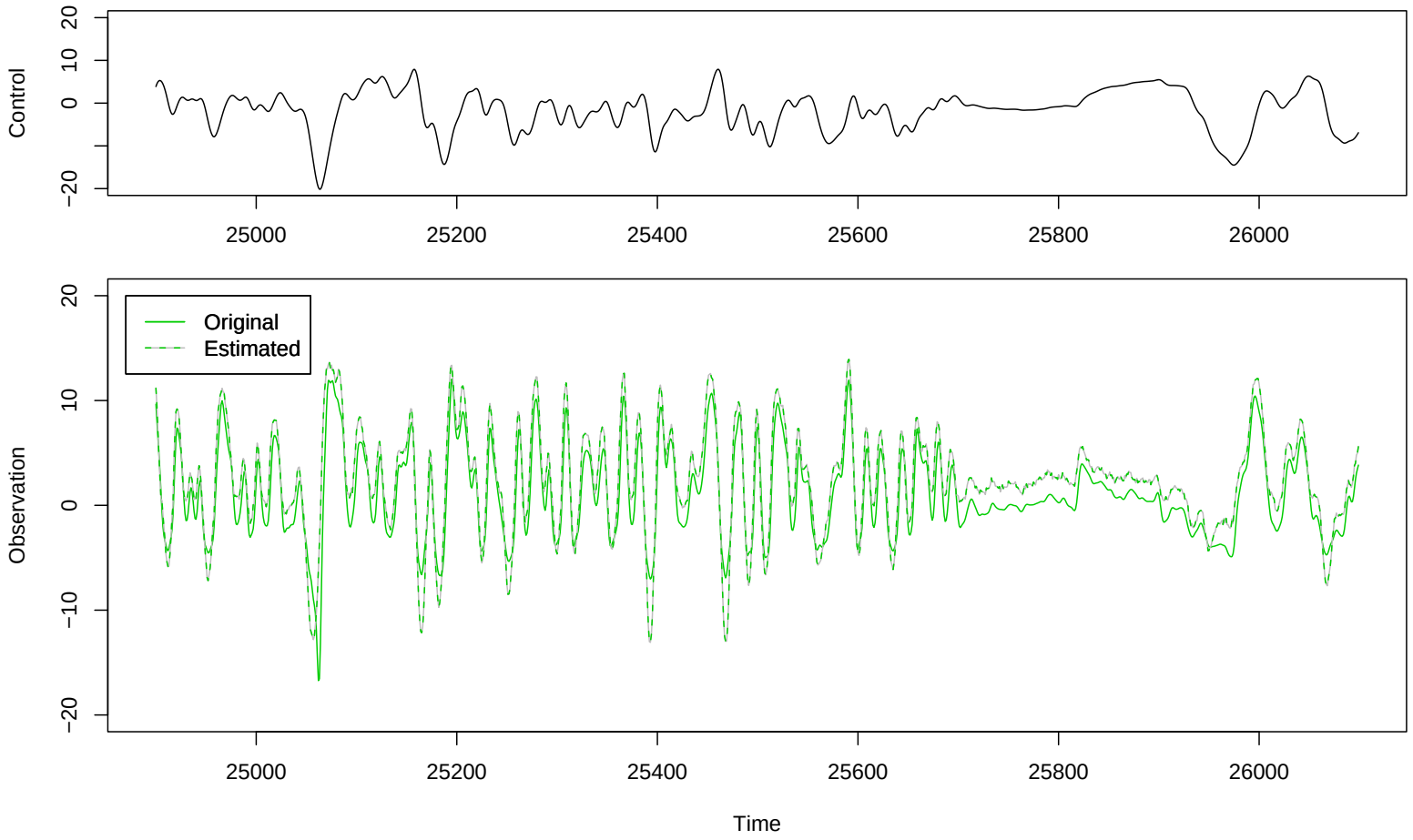


Figure 5.24: Measured and estimated output signals of data used for testing: solid green: measured output; dashed green: simulated output of the estimated model; black curve: input signal; shown are 1200 time steps of the road test data, representative for the complete sequence

and more exact models (in comparison to the original algorithm). If the local model networks were seen as neural networks, then the gradient-based adaptation would be regarded as the main part of the algorithm; the prepended division algorithm (i.e. the original LOLIMOT algorithm) would serve only to the production of "good" initial values for the parameters. At the same time, the question of the selection of the model order is answered: By applying the usual methods (validation by a separate data set, Akaike criterion etc.) the model order can be determined similar to the usual neural networks or linear system identification. It may be useful to replace the original weight functions of the local model networks. These original weight functions are normalized Gaussian bell functions, i.e. derived from the density function of the Gaussian normal distribution. These are the functions used with RBF (Radial Basis Function) networks. They may be replaced by weight functions that are a (multiplicative) superposition of sigmoidal functions, although the usage of sigmoidal functions is not really necessary; important is the derivation from an underlying decision tree, which guarantees that the weight functions are normalized. These sigmoidal functions are the usual activation functions of MLP (Multi Layer Perceptron) networks. The local model networks come thus nearer to this kind of neural networks. Only the introduction of these new weight functions gives the flexibility to add the improvements of the algorithm proposed in this thesis. Therefore, these weight functions are to be preferred.

Internal-dynamics models This especially in the nonlinear case essentially larger class compared to the models with external dynamics can be treated adequately only with the new model class. The state space dynamics which is characterized by hidden (not observable/measurable) variables (states) inside the system introduces new difficulties when identifying such systems, especially in the nonlinear case. Then new effects appear, the so-called hard nonlinearities, where hysteresis is an example. Those hard nonlinearities cannot be treated by means of linearizations, because if doing so this would completely extinct the effect. Thus, we must provide new model classes and correspondingly new identification algorithms. One approach is the new model class and identification scheme introduced here, whereas one cannot avoid the disadvantage that the computer resources which are needed for identification are multiples of those needed for models with external dynamics. This is owing to the higher flexibility of the internal-dynamics models compared to the external-dynamics models.

Grey-box approaches

The Bayes approach used for the newly introduced model class enables the inclusion of knowledge about the real system through a-priori distributions for parameters and data. Looking at the model class from this standpoint, it can equally well be seen as a grey-box approach. If one does not have such kind of knowledge, one still has the possibility to use so-called non-informative priors instead. In many cases, this choice is equivalent to the frequentist approach, but for example in the case of multiplicative parameters (like scaling parameters) there are significant differences. Also in this case, the Bayes approach shows its superiority, even if no a-priori knowledge can be used.

5.5.2 Future work

We just mention a few points for improvements of the proposed algorithm:

- Fine tuning of algorithm;
- Inclusion of choice of wavelet packet during estimation;
- Priors of wavelet coefficients with spatial dependencies in one level and dependencies through levels;
- Inclusion of other levels than the highest one in the identification;
- Combination of the algorithm with a constructive tree for spatial dependencies of wavelet coefficients (like LOLIMOT);
- Theoretic investigations on approximation properties of dynamical systems and connections to approximation of underlying functions;
- Improved particle filters;
- Adaptive estimation schemes.

5.5.3 Conclusion

We will conclude this thesis by roughly outlining a programme. Identification of nonlinear dynamical systems is an issue of high importance for modelling and simulation purposes as well for model reduction. Our programme includes the known three steps:

(i) **Models**: One must provide suitable classes of models. It should be able to include as much prior knowledge as possible in the selection of the model classes, be it in terms of smoothness or in terms of prior distributions, or, even more interesting a coupling of both (an example being the priors on wavelet coefficients and the corresponding Besov space parameters).

(ii) **Algorithms**: There must be an approximation scheme with known properties in terms of approximation rates for the selected model classes. As uncertainties are unavoidable, this approximation scheme must be coupled with a stochastic estimation procedure which deals with this uncertainties, may they originate from measurement errors, process noise or unknown states.

(iii) **Implementation**: There should be a fast implementation. Monte Carlo methods in connection with wavelet filters seem to be very promising, because they are easily parallelizable and break the curse of dimensionality.

We could hardly start with this programme in the present thesis. We just collected some theories which the author considers necessary. Many aspects equally worth to be considered have not even been mentioned, as operator theory or the theory of dynamical systems. The proposed algorithm is only a rude ad-hoc trial and certainly must be improved in many respects.

Appendix: Basic notions

Measures

We follow mainly Bauer [1992].

Signed measures

Let Ω be a set, $\mathscr{A}$ a σ-algebra on Ω and μ a measure on $\mathscr{A}$. Then we call the pair $(\Omega, \mathscr{A})$ a ***measurable space*** and the triple $(\Omega, \mathscr{A}, \mu)$ a ***measure space***. We usually assume that μ is σ-finite and non-negative. We as well consider ***signed measures***, i.e. measures which take values in

$$\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\},$$

but at most one of the values $-\infty$ or $+\infty$ is taken by μ. Each signed measure μ can be uniquely decomposed into two non-negative measures μ_+ and μ_- such that

$$\mu = \mu_+ - \mu_-.$$

Continuous and singular measures

Definition 1 (μ-continuity, μ-singularity)**:** *Let μ, ν be two (non-negative or signed) measures on the measurable space $(\Omega, \mathscr{A})$.*

(1) *ν is called μ-**continuous** if every μ-null set in $\mathscr{A}$ is also a ν-null set. We then write:*

$$\nu \ll \mu$$

(2) *ν is called μ-**singular** if there exists a set $N \in \mathscr{A}$ with $\mu(N) = 0$ and $\nu(\complement N) = 0$. We then write:*

$$\nu \perp \mu$$

Sometimes the notion "absolutely continuous" is used instead of "continuous".

Theorem 2 (Lebesgue's decomposition theorem)**:** *Let μ and ν be σ-finite measures on the measurable space $(\Omega, \mathscr{A})$. Then there exist uniquely defined measures ν_1, ν_2 on $\mathscr{A}$ with $\nu = \nu_1 + \nu_2$ such that*

$$\nu_1 \ll \mu \qquad \textit{and} \qquad \nu_2 \perp \mu.$$

Measures and densities

Definition 2: *Let $(\Omega, \mathscr{A})$ be a measurable space and $f : \Omega \longrightarrow \mathbb{R}_0^+ \cup \{\infty\}$ be an $\mathscr{A}$-measurable function. Then:*

$$\nu(A) := \int_A f d\mu \qquad A \in \mathscr{A}$$

*defines a measure and f is called the **density** (or **Radon-Nikodym derivative**) of ν with respect to μ.*

Theorem 3 (Radon-Nikodym)**:** *Let μ and ν be measures on the measurable space $(\Omega, \mathscr{A})$. If μ is σ-finite, then the following conditions are equivalent:*

(i) *ν has a density with respect to μ.*

(ii) *ν is μ-continuous.*

Theorem 4: *Let $(\Omega, \mathscr{A})$ be a measurable space and $\nu = f\mu$ a measure on $\mathscr{A}$ with density f with respect to the σ-finite measure μ on $\mathscr{A}$. Then:*

(i) *f is μ-a.e. uniquely defined.*

(ii) *ν is σ-finite if and only if f is μ-a.e. real valued.*

Function Spaces

The following is taken from Wloka [1982] and Walz [2000-2003].

Spaces of derivable functions Let Ω be an open set. Then $C^l(\Omega)$, $l \in \mathbb{N}$, is the space of real or complex valued functions $\varphi(x)$, $x \in \Omega$, with continuous and bounded derivatives

$$D^s\varphi(x), \qquad |s| \leq l$$

(up to order l). The norm on $C^l(\Omega)$ is defined by

$$\|\varphi\|_{C^l(\Omega)} := \sup_{\substack{|s|\leq l \\ x\in\Omega}} |D^s\varphi(x)|.$$

Lipschitz and Hölder spaces Let Ω be an open subset of $\mathbb{R}^d$. We say that a (real or complex valued) function φ on Ω is ***α-Hölder continuous***, if ($|\cdot|$ denoting the Euclidean norm):

$$\frac{|\varphi(x) - \varphi(y)|}{|x-y|^\alpha} \leq C < \infty$$

for all $x, y \in \Omega$, $x \neq y$, with $0 < \alpha \leq 1$ (α-Hölder continuous functions with $\alpha > 1$ are constant on Ω). If $\alpha = 1$, then we call the functions also ***Lipschitz-continuous***. If $\alpha = 0$, the function

is simply bounded. We define the space $C^{l,\alpha}$ to be the set of all functions on Ω with continuous and bounded derivatives up to order l, where additionally the l-th derivative is α-Hölder continuous. The norm of this space is given by

$$\|\varphi\|_{l,\alpha} := \sup_{\substack{|s|\le l \\ x\in\Omega}} |D^s\varphi(x)| + \sup_{\substack{|s|=l \\ x,y\in\Omega \\ x\neq y}} \frac{|D^s\varphi(x) - D^s\varphi(y)|}{|x-y|^\alpha}.$$

We have

$$C^{l,0}(\Omega) = C^l(\Omega).$$

The space $C^0(\Omega)$ is just the space of continuous and bounded functions.

A generalization of the above constructions of Lipschitz and Hölder continuous spaces is the following: Let (M,d) be a metric space. The space $\mathrm{Lip}(M,d)$ of Lipschitz continuous functions is then the space of all functions f on M such that

$$\|f\|_{\mathrm{Lip}} = \sup_{x\neq y} \frac{|f(x)-f(y)|}{d(x,y)} + |f(x_0)| < \infty$$

with some fixed point $x_0 \in M$ (if M is compact, replacing $|f(x_0)|$ by $\|f\|_\infty$ leads to an equivalent norm). $\mathrm{Lip}(M,d)$ with this norm is then a Banach space. A special case is given if $M \subseteq \mathbb{R}^d$ and $0 < \alpha \le 1$, and if d_α is the metric

$$d_\alpha(x,y) = |x-y|^\alpha.$$

The space $\mathrm{Lip}(M,d_\alpha)$ is then the Hölder space $C^{0,\alpha}(M)$.

Lipschitz domain

A Lipschitz domain is an open and bounded subset of $\mathbb{R}^d$ such that the boundary can be thought of as being locally the graph of a Lipschitz function:

Definition 3: *Let Ω be an open and bounded subset of $\mathbb{R}^d$, $d \in \mathbb{N}$, $d \ge 1$. Let $\partial\Omega$ denote the boundary of Ω. Then Ω is called **Lipschitz domain** and $\partial\Omega$ is called a **Lipschitz boundary**, if for every point $p \in \partial\Omega$, there exists a ball $B_{<r}(p) = \{x \in \mathbb{R}^d \mid \|x-p\| < r\}$ around p with radius $r > 0$ and a map*

$$h_p : B_{<r}(p) \longrightarrow B_{<1}(0)$$

on the unit ball $B_{<1}(0)$, such that

- *h_p is a bijection;*
- *h_p and h_p^{-1} are Lipschitz continuous functions;*
- *$h_p(\partial\Omega \cap B_{<r}(p)) = \{(x_1,\ldots,x_n) \in B_{<1}(0) \mid x_n = 0\}$;*
- *$h_p(\Omega \cap B_{<r}(p)) = \{(x_1,\ldots,x_n) \in B_{<1}(0) \mid x_n > 0\}$.*

Real variable Hardy spaces

The following is taken from "Encyclopeadia of Mathematics", eom.springer.de/H/h110090.htm (G.B. Folland):

The real-variable Hardy spaces $H^p = H^p(\mathbb{R}^n)$ for $0 < p < \infty$ are spaces of distributions on $\mathbb{R}^n$. Originally, they were defined as boundary values of holomorphic functions of complex Hardy spaces.

Definition Let $\phi \in \mathscr{S}(\mathbb{R}^n)$ be an element of the Schwartz class of rapidly decreasing functions, and set for all $t > 0$:

$$\phi_t(x) := t^{-n}\phi(t^{-1}x).$$

For all $f \in \mathscr{S}'(\mathbb{R}^n)$, the space of tempered distributions, define the ***radial maximal function*** $m_\phi f$ by

$$m_\phi f(x) = \sup_{t>0} |f * \phi_t(x)|$$

and the ***non-tangential maximal function*** $M_\phi f$ by

$$M_\phi f(x) = \sup_{|y-x|<t<\infty} |f * \phi_t(y)|.$$

The Fefferman-Stein theorem states that the following conditions are equivalent for $0 < p < \infty$:

1. $m_\phi f \in L^p$ for some $\phi \in \mathscr{S}$ with $\int \phi \neq 0$;
2. $M_\phi f \in L^p$ for some $\phi \in \mathscr{S}$ with $\int \phi \neq 0$;
3. $M_\phi f \in L^p$ for every $\phi \in \mathscr{S}$ with $\int \phi \neq 0$, and in fact $M_\phi f \in L^p$ uniformly for ϕ in a suitable bounded subset of $\mathscr{S}$.

The Hardy space $H^p(\mathbb{R}^n)$ is then the space of all $f \in \mathscr{S}'$ that satisfy these conditions. The quasi-norm of $H^p(\mathbb{R}^n)$ is defined to be $f \to (\int |m_\phi f|^p)^{1/p}$ (or $f \to (\int |M_\phi f|^p)^{1/p}$), different choices of ϕ leading to equivalent quasi-norms. It is a norm only in the cases $p \geq 1$, the $H^p(\mathbb{R}^n)$ in these cases being Banach spaces. Nevertheless, for $p < 1$ the p-th power $\|\cdot\|^p_{H^p(\mathbb{R}^n)}$ defines a metric that makes $H^p(\mathbb{R}^n)$ into a complete metric spaces (not Banach spaces).

Connection to L^p spaces and dual spaces For $p > 1$, the space H^p coincides with L^p, whereas H^1 is a proper subspace of L^1. For $p < 1$, H^p contains distributions that are not functions. The Fefferman theorem states that the dual of H^1 is the ***space of functions of bounded mean oscillation,*** *BMO*: It is the space of all locally integrable functions $f \in L^1_{\mathrm{loc}}(\mathbb{R}^n)$ such that

$$\|f\|_* := \sup_Q \frac{1}{|Q|} \int_Q |f(t) - f_Q| dt < \infty,$$

where the supremum is taken over all balls Q in $\mathbb{R}^n$, with volume denoted by $|Q|$, and f_Q is defined as the mean of f over Q:

$$f_Q := \frac{1}{|Q|} \int_Q f(t) dt.$$

The so-called *BMO*-norm $\|\cdot\|_*$ becomes a norm after dividing *BMO* by the constant functions.

The dual space of H^p for $p < 1$ is the homogeneous Lipschitz space of order $n(1/p-1)$.

The spaces H^p for $(p \leq 1)$ and *BMO* have more desirable properties than the corresponding L^p spaces, and thus provide an extension to the scale of L^p spaces for $1 < p < \infty$ that is in many respects more natural and useful than the L^p spaces for $p \leq 1$ and $p = \infty$, respectively. Examples are Calderón-Zygmund operators and Littlewood-Paley theory.

Atomic decompositons In the case $p \leq 1$, there is an interesting and important characterization of the Hardy spaces $H^p(\mathbb{R}^n)$ by atomic decompositions: A measurable function α is called a p-atom for $0 < p \leq 1$ if

(1) α vanishes outside some ball of radius $r > 0$, and

$$\sup_{x \in \mathbb{R}^n} |\alpha(x)| \leq r^{-n/p},$$

(2) for all polynomials P of degree $\leq n(1/p-1)$ we have

$$\int P(x)\alpha(x)dx = 0.$$

The atomic decomposition theorem states that $f \in H^p(\mathbb{R}^n)$ if and only if

$$f = \sum_{j \in \mathbb{N}} c_j \alpha_j,$$

where each α_j is a p-atom and

$$\sum_{j \in \mathbb{N}} |c_j|^p < \infty.$$

Hilbert spaces

Riesz basis of a Hilbert space The following can be found for example in Jaffard et al. [2001].

Definition 4: *A **Riesz Basis** of a Hilbert space H is the image of a Hilbert basis $(f_j)_{j \in J}$ of H under an isomorphism $T : H \longrightarrow H$*

Note: T is not necessarily an isometry!
Then: Each $x \in H$ is decomposed uniquely in a series

$$x = \sum_{j \in J} \alpha_j e_j \quad \text{where} \sum_{j \in J} |\alpha_j|^2 < \infty$$

Furthermore:

$$\alpha_j = \langle x, e_j^* \rangle \quad \text{where } e_j^* := (T^*)^{-1}(f_j)$$

$(e_j^*)_{j \in J}$ is the ***dual basis*** of $(e_j)_{j \in J}$; the two systems are said to be ***biorthogonal***.

Fourier transform

We follow Walz [2000-2003].

Fourier transform on the Schwartz space

Let $\mathscr{S}(\mathbb{R}^n)$ be the Schwartz space of smooth functions with rapid decrease. The ***Fourier transformed function***

$$\hat{f} : \mathbb{R}^n \longrightarrow \mathbb{C}$$

of a function $f \in \mathscr{S}(\mathbb{R}^n)$ is given by

$$\hat{f}(\xi) := \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} f(x) e^{-ix\cdot\xi} d^n x$$

with $x \cdot \xi := x_1\xi_1 + \cdots + x_n\xi_n$. We have then also $\hat{f} \in \mathscr{S}(\mathbb{R}^n)$, and the following fundamental properties:

(i) For every multiindex $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, setting

$$D^\alpha f := \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \cdots \frac{\partial^{\alpha_n}}{\partial x_n^{\alpha_n}} f$$

and

$$(ix)^\alpha := (ix_1)^{\alpha_1} \cdots (ix_n)^{\alpha_n},$$

we have

$$\widehat{D^\alpha f} = (ix)^\alpha \hat{f}.$$

(ii) For every $g \in \mathscr{S}$ and the convolution product

$$f * g(x) := \int_{\mathbb{R}^n} f(x-y) g(y) d^n y,$$

we have

$$\widehat{f * g} = (2\pi)^{n/2} \hat{f}\hat{g}.$$

(iii) We have:

$$\int_{\mathbb{R}} |f(x)|^2 d^n x = \int_{\mathbb{R}^n} |\hat{f}(x)|^2 d^n x.$$

The ***Fourier transform*** $\mathscr{F}$ is the linear and bijective map

$$\mathscr{F} : \mathscr{S}(\mathbb{R}^n) \longrightarrow \mathscr{S}(\mathbb{R}^n), \qquad \mathscr{F} f = \hat{f}.$$

The inverse map $\mathscr{F}^{-1}$ is called ***inverse Fourier transform***. For $\tilde{f} := \mathscr{F}^{-1} f$ the following inversion formula is valid:

$$\tilde{f}(\xi) = \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} f(x) e^{ix\cdot\xi} d^n x$$

(where the only difference to the Fourier transform $\mathscr{F}$ is the sign in the exponent of the *e*-function).

Fourier transform on L^p spaces

With the same formulas, the Fourier transform can be defined for L^p spaces. For $f \in L^1$, the ***lemma of Riemann-Lebesgue*** says that $\hat{f} \in C_0^0(\mathbb{R}^n)$, i.e. $\hat{f}$ is continuous and

$$\lim_{|\xi|\to\infty} \hat{f}(\xi) = 0.$$

For $f \in L^1 \cap L^p$, $1 \leq p \leq 2$, and q with

$$1/p + 1/q = 1,$$

the ***Hausdorff-Young inequality*** is valid:

$$\|\mathscr{F} f\|_q \leq \|f\|_p,$$

and the Fourier transform can thus be extended uniquely to a continuous linear map from $L^p(\mathbb{R}^n)$ to $L^q(\mathbb{R}^n)$. This extension is as well called Fourier transform. In the case of the Hilbert space L^2, this extension $\mathscr{F}$ is a unitary operator, especially we have the ***theorem of Plancherel***:

$$\|\mathscr{F} f\|_2 = \|f\|_2 \qquad \text{for } f \in L^2.$$

Fourier transform for finite measures

The Fourier transform can also be defined for a finite measure μ on the Borel-σ-algebra $\mathfrak{B}(\mathbb{R}^n)$. The ***Fourier transformed measure*** $\hat{\mu}$ is defined as

$$\hat{\mu} : \mathbb{R}^n \longrightarrow \mathbb{C}, \qquad \hat{\mu}(\xi) := \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} f(x) e^{-ix\cdot\xi} d^n x.$$

For arbitrary finite Borel measures μ, ν and $\alpha \in \mathbb{R}_{\geq 0}$, the following properties hold:

(i) $\widehat{\mu + \nu} = \hat{\mu} + \hat{\nu}$,

(ii) $\widehat{\alpha\mu} = \alpha\hat{\mu}$,

(iii) $\widehat{\mu * \nu} = \hat{\mu} \cdot \hat{\nu}$.

If further T is a linear map on $\mathbb{R}^n$ into itself with transposed map $T^\top$, then we have for the measure $T(\mu)$:

$$\widehat{T(\mu)} = \hat{\mu} \circ T^\top.$$

Especially for the translation $T_\alpha(x) := x + \alpha$ with $\alpha \in \mathbb{R}^n$, we have

$$\widehat{T_\alpha(\mu)} = \hat{\delta}_\alpha \hat{\mu}$$

where δ_α is the Dirac measure in α.

Sequence Space $\ell^2(\mathbb{Z})$

We follow Jaffard et al. [2001].

Convolution on $\ell^2(\mathbb{Z})$

Definition 5: *For each $g,h \in \ell^2(\mathbb{Z})$ we define the convolution $*$ by*

$$(g*h)_n := \sum_{k\in\mathbb{Z}} g_k h_{n-k} \quad \text{for } g=(g_k)_{k\in\mathbb{Z}}, h=(h_k)_{k\in\mathbb{Z}} \in \ell^2(\mathbb{Z}).$$

The convolution operator $*$ has the following properties:

- $*$ is associative, distributive with respect to addition, and commutative.
- $(g*h)_n = \sum_{k\in\mathbb{Z}} g_k h_{n-k} = \sum_{k\in\mathbb{Z}} g_{n-k} h_k$.
- $(g*e_j)_n = \sum_{k\in\mathbb{Z}} g_{n-k}\delta_{k,j} = g_{n-j}$ (g is shifted by j to the right).

Shifts on $\ell^2(\mathbb{Z})$

Definition 6: *Let $n \in \mathbb{Z}$. We define the **shift (translation)** $\Psi_n : \ell^2 \longrightarrow \ell^2$ by:*

$$(\Psi_n h)_k := h_{k-n} \quad \text{for } h=(h_k)_{k\in\mathbb{Z}} \in \ell^2(\mathbb{Z}).$$

Properties:

- Ψ_n is an isometry: $\|\Psi_n g\|_{\ell^2} = \|g\|_{\ell^2}$.
- $\Psi_n \Psi_m = \Psi_{n+m}$ for all $n,m \in \mathbb{Z}$.

Definition 7: *Let T be an operator on $\ell^2(\mathbb{Z})$. We call T **translation invariant** if*

$$\Psi_n T = T\Psi_n \qquad \text{for all shifts } \Psi_n,\ n\in\mathbb{Z},$$

i.e. T commutes with Ψ_n.

It suffices $\Psi_1 T = T\Psi_1$.

Translation invariant linear bounded operators on $\ell^2(\mathbb{Z})$

Theorem 5: *Let $F : \ell^2(\mathbb{Z}) \longrightarrow \ell^2(\mathbb{Z})$ be a linear bounded operator which is translation invariant, i.e. for all $u=(u_k)_{k\in\mathbb{Z}} \in \ell^2(\mathbb{Z})$ and $n \in \mathbb{Z}$ holds:*

$$(F\tilde{u})_{k+n} = (Fu)_k \qquad \text{where } (\tilde{u}_k)_{k\in\mathbb{Z}} \in \ell^2(\mathbb{Z}) \text{ with } \tilde{u}_{k+n} := u_k.$$

Then there exists a sequence $h=(h_k)_{k\in\mathbb{Z}} \in \ell^2(\mathbb{Z})$, such that

$$Fu = h*u \qquad \text{i.e.} \qquad (Fu)_k = \sum_{n\in\mathbb{Z}} h_{k-n}u_n \qquad \text{for} \qquad u=(u_k)_{k\in\mathbb{Z}} \in \ell^2(\mathbb{Z}).$$

Furthermore, $H(\omega) := \sum_{k\in\mathbb{Z}} h_k e^{-ik\omega}$ is in $L^\infty(0,2\pi)$ with norm $\|H\|_\infty = \|F\|$.
Conversely, if $h=(h_k)_{k\in\mathbb{Z}} \in \ell^2(\mathbb{Z})$ is such that $H \in L^\infty(0,2\pi)$, then $Fu := h*u$ defines a linear bounded translation invariant operator $F : \ell^2(\mathbb{Z}) \longrightarrow \ell^2(\mathbb{Z})$ with norm $\|F\| = \|H\|_\infty$

Neural Networks

Kolmogorov's Neural Net

We follow Walz [2000-2003].

The 13th problem of Hilbert states as follows: Is it possible to realize any continuous m-dimensional function on a compact n-dimensional set by superposition and composition of one-dimensional continuous functions? Kolmogorov's neural net (1957) as solution to this problem was constructed only for theoretical reasons:

Theorem 6: *Let $K \subset \mathbb{R}^n$, $K \neq \emptyset$, a compact subset of $\mathbb{R}^n$ and $f : K \longrightarrow \mathbb{R}^m$ a continuous (vector-valued) function $f = (f_1, \ldots, f_m)$. Then there exist continuous functions $\varphi_{ip} : \mathbb{R} \longrightarrow \mathbb{R}$, $1 \leq i \leq n$, $1 \leq p \leq 2n+1$, and $T_j : \mathbb{R} \longrightarrow \mathbb{R}$, $1 \leq j \leq m$, such that for all $x = (x_1, \ldots, x_n) \in K$ and all $j \in \{1, \ldots, m\}$ holds*

$$f_j(x) = \sum_{p=1}^{2n+1} T_j\Big(\sum_{i=1}^{n} \varphi_{ip}(x_i) \Big).$$

The practical realization of Kolmogorov's net was not successful because of the complicated dependence of the T_j on f_j.

At the end of the 1980s Robert Hecht-Nielsen proposed the ridge-type neural network; following his proposal the following result was found:

Theorem 7: *Let $K \subset \mathbb{R}^n$, $K \neq \emptyset$, a compact subset of $\mathbb{R}^n$ and $f : K \longrightarrow \mathbb{R}^m$ a continuous (vector-valued) function $f = (f_1, \ldots, f_m)$. Then there exist for all $\varepsilon > 0$ and all continuous sigmoidal transfer functions $T : \mathbb{R} \longrightarrow \mathbb{R}$ net parameters $q \in \mathbb{N}$, $w_{ip}, \Theta_p, g_{pj} \in \mathbb{R}$, $1 \leq i \leq n$, $1 \leq p \leq q$, $1 \leq j \leq m$, such that for all $x = (x_1, \ldots, x_n) \in K$ and all $j \in \{1, \ldots, m\}$ holds*

$$\Big| f_j(x) - \sum_{p=1}^{q} g_{pj} T\Big(\sum_{i=1}^{n} w_{ip} x_i - \Theta_p \Big) \Big| \leq \varepsilon.$$

Bibliography

J. Abonyi and R. Babuška. Local and global identification and interpretation of parameters in Takagi-Sugeno fuzzy models. In *FUZZ-IEEE'00 Conference, Arizona, USA*, pages 835–840, 2000.

J. Abonyi, R. Babuška, L.F.A. Wessels, H.B. Verbruggen, and F. Szeifert. Fuzzy modeling and model based control with use of a priori knowledge. In *Mathmod 2000*, pages 769–772, 2000a.

Janos Abonyi, R. Babuška, Lajos Nagy, and Ferenc Szeifert. Local and Global Identification for Fuzzy Model Based Control. In *Proceedings of the Intelligent Systems in Control and Measurement Symposium, INTCOM 2000, Veszprem, Hungary*, pages 111–116, 2000b.

Janos Abonyi, Tibor Chovan, and Ferenc Szeifert. Identification of Nonlinear Systems using Gaussian Mixture of Local Models. *Hungarian Journal of Industrial Chemistry*, 29(2): 129–135, 2001.

F. Abramovich and Y. Benjamini. Adaptive thresholding of wavelet coefficients. *Computational Statistics and Data Analysis*, 22:351–361, 1996.

F. Abramovich and Y. Benjamini. Thresholding of wavelet coefficients as multiple hypothesis testing procedure. In *A. Antoniadis and G. Oppenheim (eds.)*, Wavelets and Statistics, pages 5–14. Springer-Verlag, New York, 1995.

F. Abramovich, T. Sapatinas, and B.W. Silverman. Wavelet Thresholding via a Bayesian Approach. *Journal of the Royal Statistical Society, Series B*, 60:725–749, 1998.

J. Aczél. *Lectures on Functional Equations and Their Applications*. Academic Press, New York/London, 1966.

J.H. Ahrens and U. Dieter. Computer Generation of Poisson Deviates from Modified Normal Distributions. *ACM Transactions on Mathematical Software*, 8(2):163–179, 1982.

J.H. Ahrens and U. Dieter. Computer Methods for Sampling from the Exponential and Normal Distributions. *Communications of the ACM*, 15(10):873–882, 1972.

J.H. Ahrens, K.D. Kohrt, and U. Dieter. ALGORITHM 599 Sampling from Gamma and Poisson Distributions. *ACM Transactions on Mathematical Software*, 9(2):255–257, 1983.

H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Proceedings of the 2nd International Symposium on Information Theory (B.N. Petrov and F. Csaki eds.), Akademiai Kiado, Budapest*, pages 267–281, 1973.

H. Akashi and H. Kumamoto. Construction of discrete-time nonlinear filter by Monte Carlo methods with variance-reducing techniques (in Japanese). *Systems and Control*, 19(4):211–221, 1975.

B.D.O. Anderson and J.B. Moore. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, 1979.

S. Arnborg and G. Sjödin. On the foundations of Bayesianism. In *MaxEnt 2000, Proceedings of the Twentieth International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Gif-surYvette. American Institute of Physics*, pages 61–71, 2001.

Stefan Arnborg. Robust Bayesianism: Imprecise and Paradoxical Reasoning. In *Proceedings of the Seventh International Conference on Information Fusion, International Society of Information Fusion. Mountain View, CA.*, volume 1, pages 407–414, 2004.

Stefan Arnborg. Robust Bayesianism: Relation to Evidence Theory. *ISIF Journal of Advances in Information Fusion*, 1(1):75–90, 2006.

Stefan Arnborg and Gunnar Sjödin. Bayes Rules in Finite Models. In *ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, Germany. IOS Press*, pages 571–575, 2000.

Stefan Arnborg and Gunnar Sjödin. What is the plausibility of probability? (revised 2003). Manuscript, ftp.nada.kth.se/pub/documents/Theory/Stefan-Arnborg/m2001.pdf, 2003.

M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

Krishna B. Athreya, Hani Doss, and Jayaram Sethuraman. On the convergence of the Markov chain simulation method. *The Annals of Statistics*, 24(1):69–100, 1996.

R. Balian. Un principe d'incertitude fort en théorie du signal ou en mécanique quantique. *Comptes-Rendus de l'Académie des Sciences (Paris) Série II*, 292:1357–1361, 1981.

Arunava Banerjee. Initializing Neural Networks using Decision Trees. In *Proceedings of the International Workshop on Computational Learning and Natural Learning Systems*, 1994.

Stephen Banks, editor. *Signal Processing, Image Processing and Pattern Recognition*. Prentice Hall, Englewood Cliffs, 1990.

Stephen Paul Banks. *Mathematical Theories of Nonlinear Systems*. Prentice Hall International Series in Systems and Control Engineering. Prentice Hall, New York, 1988.

Ole E. Barndorff-Nielsen, David R. Cox, and Claudia Klüppelberg, editors. *Complex Stochastic Systems*, volume 87 of *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC, Boca Raton, 2001.

Eva Barrena Algara. *SODT: Soft Operator Decision Trees*. Dissertation, Technical University of Kaiserslautern, 2007.

Andrew Barron, Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Approximation and learning by greedy algorithms. *Annals of Statistics*, to appear.

Heinz Bauer, editor. *Maß- und Integrationstheorie (2., überarbeitete Auflage)*. Walter de Gruyter, Berlin, 1992.

L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

James O. Berger, editor. *Statistical Decision Theory: Foundations, Concepts and Methods*. Springer Series in Statistics. Springer-Verlag, New York, 1980.

James O. Berger and Robert L. Wolpert, editors. *The Likelihood Principle*, volume 6 of *Institute of Mathematical Statistics Lecture Notes–Monograph Series*. Institute of Mathematical Statistics, Hayward, California, 1984.

J.O. Berger and J.M. Bernardo. Estimating a product of means: Bayesian analysis with reference priors. *Journal of the American Statistical Association*, 84:200–207, 1989.

J.O. Berger and J.M. Bernardo. Ordered group reference priors with application to the multinomial problem. *Biometrika*, 79(1):25–37, 1992.

Jonathan Berger and Charles Nichols. Brahms at the Piano: An Analysis of Data from the Brahms Cylinder. *Leonardo Music Journal*, 4:23–30, 1994.

J. Bergh and J. Löfström. *Interpolation Spaces: An Introduction*. Springer, Berlin, 1976.

J.M. Bernardo. Reference posterior distributions for Bayesian inference (with discussion). *Journal of the Royal Statistical Society (Series B)*, 41:113–147, 1979.

José M. Bernardo. Bayesian Statistics. In *(R. Viertl, ed.)*, Encyclopedia of Life Support Systems (EOLSS), *Vol.* Probability and Statistics. UNESCO, Oxford, UK, 2003.

José M. Bernardo. Bayesian Statistics. In *S.N. Durlauf and L. E. Blume (eds.)*, The New Palgrave Dictionary of Economics, *2nd ed.* Palgrave Macmillan, New York, in press.

C. Berzuini, N. Best, W. Gilks, and C. Larizza. Dynamic conditional independence models and Markov chain Monte Carlo. *Journal of the American Statistical Association*, 92(440): 1403–1412, 1997.

Carlo Berzuini and Walter Gilks. RESAMPLE-MOVE Filtering with Cross-Model Jumps. In *Arnaud Doucet, Nando de Freitas and Neil Gordon (eds.)*, Sequential Monte Carlo Methods in Practice, pages 117–138. Springer Verlag, New York, 2001.

David R. Bickel. Incorporating expert knowledge into frequentist results by combining subjective prior and objective posterior distributions: A generalization of confidence distribution combination. Manuscript, arXiv.org:math.ST/0602377, 2006.

M. Birman and M. Solomyak. Piecewise polynomial approximation of functions of the classes W_p^α. *Matematicheskii Sbornik*, 73(3):331–355, 1967.

Martin Brokate and Jürgen Sprekels. *Hysteresis and Phase Transitions*. Springer-Verlag, New York, 1996.

Stephen Brooks. Markov Chain Monte Carlo method and its application. *The Statistician*, 47 (1):69–100, 1998.

Martin Burger and Heinz W. Engl. Training Neural Networks with Noisy Data as an Ill-Posed Problem. *Advances in Computational Mathematics*, 13(4):335–354, 2000. doi: 10.1023/A:1016641629556.

Martin Burger and Andreas Neubauer. Analysis of Tikhonov Regularization for Function Approximation by Neural Networks. *Neural Networks*, 16:79–90, 2003.

Martin Burger and Andreas Neubauer. Error Bounds for Approximation with Neural Networks. *Journal of Approximation Theory*, 112:235–250, 2001.

Kenneth P. Burnham and David R. Anderson. Multimodel Inference: Understanding AIC and BIC in Model Selection. *Sociological Methods & Research*, 33(2):261–304, 2004.

A.P. Calderón. Intermediate spaces and interpolation, the complex method. *Studia Mathematica*, 24:113–190, 1964.

O. Cappé, A. Guillin, J.-M. Marin, and C.P. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.

O. Cappé, S.J. Godsill, and E. Moulines. An overview of existing methods and recent advances in Sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

Olivier Cappé, Eric Moulines, and Tobias Rydén, editors. *Inference in Hidden Markov Models*. Springer-Verlag, New York, 2005.

G. Casella and C.P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1): 81–94, 1996.

F. Cérou, F. LeGland, and N. Newton. Stochastic particle methods for linear tangent filtering equations. In *J.-L. Menaldi, E. Rofman and A. Sulem (eds.)*, Optimal Control and Partial Differential Equations — Innovations and Applications, in Honor of Professor Alain Bensoussan's 60th Anniversary, pages 231–240. IOS Press, Amsterdam, 2000.

H. Chipman, E. Kolaczyk, and R. McCulloch. Adaptive Bayesian Wavelet Shrinkage. *Journal of the American Statistical Association*, 92(440):1413–1421, 1997.

Hugh A. Chipman and Lara J. Wolfson. Prior Elicitation in the Wavelet Domain. In *Peter Müller and Brani Vidakovic (eds.)*, Bayesian Inference in Wavelet-Based Models, pages 83–94. Springer Verlag, New York, 1999.

N. Chopin. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Annals of Statistics*, 32(6):2385–2411, 2004.

T.C. Clapp and S.J. Godsill. Fixed-lag smoothing using sequential importance sampling. In *J.M. Bernardo, J.O. Berger, A.P. Dawid and A.F.M. Smith (eds.)*, Bayesian Statistics, *Vol. 6*, pages 743–752. Oxford University Press, Oxford, 1999.

M.A. Clyde, G. Parmigiani, and B. Vidakovic. Multiple shrinkage and subset selection in wavelets. *Biometrika*, 85:391–401, 1998.

A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications of Pure and Applied Mathematics*, 45:485–560, 1992.

A. Cohen, I. Daubechies, and P. Vial. Wavelets on the interval and fast wavelet transforms. *Applied and Computational Harmonic Analysis*, 1:54–81, 1993.

A. Cohen, R. DeVore, and R. Hochmuth. Restricted nonlinear approximation. *Contructive Approximation*, 16:85–113, 2000.

R.T. Cox. Probability, frequency, and reasonable expectation. *American Journal of Physics*, 14:1–13, 1946.

D. Crisan and A. Doucet. A survey on convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3):736–746, 2002.

M. Crouse, R. Nowak, and R. Baraniuk. Wavelet-based statistical signal processing using Hidden Markov Models. *IEEE Transactions on Signal Processing*, 46:886–902, 1998.

W. Dahmen. Wavelets and multiscale methods for operator equations. *Acta Numerica*, 6: 55–228, 1997.

I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications of Pure and Applied Mathematics*, 41(7):909–996, 1988.

I. Daubechies. *Ten Lectures on Wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 1992.

G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Journal of Constructive Approximation*, 13:57–98, 1997.

D. de Brucq, O. Colot, and A. Sombo. Identical foundation of probability theory and fuzzy set theory. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 2, pages 1442–1449, 2002.

M. De Gunst, H.R. Künsch, and B. Schouten. Statistical analysis of ion channel data using hidden Markov models with correlated state-dependent noise and filtering. *Journal of the American Statistical Association*, 96:805–815, 2001.

P. Del Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, New York, 2004.

P. Del Moral. Measure-valued processes and interacting particle systems. Application to non-linear filtering problems. *Annals of Applied Probability*, 8(2):438–495, 1998.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo Samplers. *Journal of the Royal Statistical Society, Series B*, 68(3):411–436, 2006.

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

R. DeVore, B. Jawerth, and V. Popov. Compression of wavelet decompositions. *American Journal of Mathematics*, 114:737–785, 1992.

R. A. DeVore and R.C. Sharpley. Besov spaces on domains in $\mathbb{R}^d$. *Transactions of the American Mathematical Society*, 335(2):843–864, 1993.

R. A. DeVore and R.C. Sharpley. *Maximal Functions Measuring Smoothness*, volume 293 of *Memoirs of the American Mathematical Society*. American Mathematical Society, Providence, RI, 1984.

R.A. DeVore and V.A. Popov. Interpolation spaces and nonlinear approximation. In *M. Cwikel et al. (eds.)*, Function Spaces and Applications: Proceedings of the US-Swedish Seminar held in Lund, Sweden, 1986, *Vol. 1302 of Lecture Notes in Mathematics*, pages 191–205. Springer, Berlin, 1988.

R.A. DeVore and V. Temlyakov. Some remarks on greedy algorithms. *Advances in Computational Mathematics*, 5:173–187, 1996.

Ronald A. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.

Ronald A. DeVore and George G. Lorentz. *Constructive Approximation*, volume 303 of *Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen*. Springer, Berlin – Heidelberg, 1993.

D. Donoho, I. Johnstone, G. Kerkyacharian, and D. Picard. Wavelet Shrinkage: Asymptopia? *Journal of the Royal Statistical Society, Series B*, 57:301–369, 1995.

David L. Donoho. CART and Best-Ortho-Basis: A Connection. *Annals of Statistics*, 25(5): 1870–1911, 1997.

D.L. Donoho and I.M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90:1200–1224, 1995.

D.L. Donoho and I.M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

R. Douc, O. Cappé, and E. Moulines. Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA), Zagreb, Croatia, September 2005*, pages 64–69, 2005.

A. Doucet. *Monte Carlo methods for Bayesian estimation of hidden Markov models. Application to radiation signals (in French)*. Ph.d. thesis, University Paris-Sud Orsay, 1997.

A. Doucet. On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR.310, University of Cambridge, Department of Engineering, 1998.

A. Doucet and V.B. Tadić. Parameter estimation in general state-space models using particle methods. *Annals of the Institute of Mathematical Statistics*, 55(2):409–422, 2003.

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

Arnaud Doucet, Nando de Freitas, and Neil Gordon. An Introduction to Sequential Monte Carlo Methods. In *Arnaud Doucet, Nando de Freitas and Neil Gordon (eds.)*, Sequential Monte Carlo Methods in Practice, pages 3–14. Springer Verlag, New York, 2001a.

Arnaud Doucet, Nando de Freitas, and Neil Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer Verlag, New York, 2001b.

Gérard Dreyfus and Yizhak Idan. The canonical form of nonlinear discrete-time models. *Neural Computation*, 10:133–164, 1998.

Nelson Dunford and Jacob T. Schwartz, editors. *Linear Operators. Part I: General Theory*. Interscience Publishers, New York, 1957.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic models of proteins and nucleid acids*. Cambridge University Press, Cambridge, 1998.

N. Dyn, D. Leviatan, D. Levin, and A. Pinkus, editors. *Multivariate Approximation and Applications*. Cambridge University Press, Cambridge, 2001.

R.G. Edwards and A.D. Sokal. Generalization of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm. *Physical Review, D*, 38(6):2009–2012, 1988.

Heinz W. Engl, Martin Hanke, and Andreas Neubauer, editors. *Regularization of Inverse Problems*. Kluwer Academic Publishers, Dordrecht / Boston / London, 2000.

P. Fearnhead. Computational methods for complex stochastic systems: A review of some alternatives to MCMC. *Statistics and Computing*, in press.

P. Fearnhead. Markov chain Monte Carlo, sufficient statistics and particle filters. *Journal of Computational and Graphical Statistics*, 11(4):848–862, 2002.

Arie Feuer, Paul M.J. Van den Hof, and Peter S.C. Heuberger. A Unified Transform for LTI Systems—Presented as a (Generalized) Frame. *EURASIP Journal on Applied Signal Processing*, pages 1–9, 2006. doi: 10.1155/ASP/2006/91604.

Mário A.T. Figueiredo and Robert D. Nowak. Wavelet-Based Image Estimation: An Empirical Bayes Approach Using Jeffreys' Noninformative Prior. *IEEE Transactions on Image Processing*, 10(9):1322–1331, 2001.

Dean P. Foster and Robert A. Stine. An Information Theoretic Comparison of Model Selection Criteria. IDEAS, RePEc Handle: RePEc:nwu:cmsems:1180, 1997.

D.P. Foster and E.I. George. The Risk Inflation Criterion for multiple regression. *Annals of Statistics*, 22(4):1947–1975, 1994.

Dieter Fox, Sebastian Thrun, Wolfram Burgard, and Frank Dellaert. Particle Filters for Mobile Robot Localization. In *Arnaud Doucet, Nando de Freitas and Neil Gordon (eds.),* Sequential Monte Carlo Methods in Practice, pages 401–428. Springer Verlag, New York, 2001.

H. Gao. Wavelet shrinkage denoising using nonnegative garrote. *Journal of Computational and Graphical Statistics*, 7(4):469–488, 1998.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

W.R. Gilks and C. Berzuini. Following a moving target — Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society, Series B*, 63(1):127–146, 2001.

S. Godsill and T. Clapp. Improvement strategies for Monte Carlo particle filters. In *Arnaud Doucet, Nando de Freitas and Neil Gordon (eds.),* Sequential Monte Carlo Methods in Practice, pages 139–158. Springer Verlag, New York, 2001.

S. Godsill and P. Rayner. *Digital Audio Restoration: A statistical model-based approach*. Springer, Berlin, 1998.

N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE-Proceedings-F*, 140(2):107–113, 1993.

Peter J. Green. A Primer on Markov Chain Monte Carlo. In *Ole E. Barndorff-Nielsen, David R. Cox and Claudia Klüppelberg (eds.),* Complex Stochastic Systems, pages 1–62. Chapman & Hall/CRC, Boca Raton, 2001.

Peter J. Green, Nils Lid Hjort, and Sylvia Richardson, editors. *Highly Structured Stochastic Systems*, volume 27 of *Oxford Statistical Science Series*. Oxford University Press, USA, 2003.

P.J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

Charles W. Groetsch, editor. *Inverse Problems in the Mathematical Sciences*. Vieweg, Braunschweig Wiesbaden, 1993.

Dong Guo, Xiaodong Wang, and Rong Chen. Wavelet-Based Sequential Monte Carlo Blind Receivers in Fading Channels With Unknown Channel Statistics. *IEEE Transactions on Signal Processing*, 52(1):227–239, 2004.

J. Halpern. A counterexample to theorems of Cox and Fine. *Journal of Artificial Intelligence Research*, 10:67–85, 1999a.

J. Halpern. Cox's theorems revisited. *Journal of Artificial Intelligence Research*, 11:429–435, 1999b.

J. Handschin. Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica*, 6:555–563, 1970.

J. Handschin and D. Mayne. Monte Carlo techniques to estimate the conditional expectation in mutli-stage non-linear filtering. *International Journal of Control*, 9(5):547–559, 1969.

Andrew Harvey, Siem Jan Koopman, and Neil Shephard, editors. *State Space and Unobserved Component Models: Theory and Applications*. Cambridge University Press, Cambridge, 2004.

Zygmunt Hasiewicz. Non-parametric estimation of non-linearity in a cascade time-series system by multiscale approximation. *Signal Processing*, 81(4):791–807, 2001.

T.J. Hastie and R.J. Tibshirani. *Generalized Additive Models*. Chapman and Hall, London, 1990.

W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

A.J. Haug. A Tutorial on Bayesian Estimation and Tracking Techniques Applicable to Nonlinear and Non-Gaussian Processes. Technical report, MITRE Corporation, 2005.

Peter S.C. Heuberger, Thomas J. De Hoog, Paul M.J. van den Hof, and Bo Wahlberg. Orthonormal basis functions in time and frequency domain: Hambo transform theory. *SIAM Journal of Control and Optimization*, 42(4):1347–1373, 2003.

Harro Heuser, editor. *Gewöhnliche Differentialgleichungen: Einführung in Lehre und Gebrauch (4. Auflage)*. B.G. Teubner Verlag, Wiesbaden, 2004.

Diederich Hinrichsen and Anthony J. Pritchard. *Mathematical Systems Theory I: Modelling, State Space Analysis, Stability and Robustness*, volume 48 of *Texts in Applied Mathematics*. Springer Verlag, Berlin Heidelberg, 2005.

E. Hlawka. Funktionen beschränkter Variation in der Theorie der Gleichverteilung. *Annali di Mathematica Pura ed Applicata*, 54:325–333, 1961.

Frank Hoffmann and Oliver Nelles. Genetic programming for model selection of TSK-fuzzy systems. *Information Sciences*, 136:7–28, 2001.

Jeroen D. Hol, Thomas B. Schön, and Fredrik Gustafsson. On resampling algorithms for particle filters. Proceedings of Nonlinear Statistical Signal Processing Workshop (NSSPW), 2006.

C.C. Holmes and D.G.T. Denison. Bayesian Wavelet Analysis with a Model Complexity Prior. In *J.M. Bernardo, J.O. Berger, A.P. Dawid and A.F.M. Smith (eds.)*, Bayesian Statistics, *Vol. 6*, pages 769–776. Oxford University Press, Oxford, 1999.

G. Huerta and M. West. Bayesian inference on periodicities and component spectral structure in time series. *Journal of Time Series Analysis*, 20(4):401–416, 1999.

J.P. Hughes, P. Guttorp, and S.P. Charles. A non-homogeneous hidden Markov model for precipitation. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 48(1): 15–30, 1999.

Don Hush. Training a Piecewise-linear Sigmoid Node is Hard. Technical Report EECE98-001 (Version 1.0), UNM, 1998.

Don Hush, Clint Scovel, and Ingo Steinwart. Stability of Unstable Learning Algorithms. Technical Report LA-UR-03-4845, LANL, 2003.

Don Hush, Clint Scovel, and Ingo Steinwart. Stability of Unstable Learning Algorithms. *Machine Learning*, 67(3):197–206, 2007.

Don R. Hush. Training a Sigmoidal Node is Hard. *Neural Computation*, 11(5):1249–1260, 1999.

Don R. Hush and Bill Horne. Efficient Algorithms for Function Approximation with Piecewise Linear Sigmoidal Networks. *IEEE Transactions on Neural Networks*, 9(6):1129–1141, 1998.

Stéphane Jaffard, Yves Meyer, and Robert D. Ryan, editors. *Wavelets: Tools for Science & Technology*. SIAM, Philadelphia, 2001.

E.T. Jaynes. Confidence Intervals vs Bayesian Intervals. In *Harper and Hooker (eds.)*, Foundations of Probability Theory, Statistical Inference, and Statistical Theories of Science, *Vol. II*, pages 175–257. Reidel Publishing Company, Dordrecht-Holland, 1976.

E.T. Jaynes. Marginalization and prior probabilities. In *A. Zellner (ed.)*, Bayesian Analysis in Econometrics and Statistics. North-Holland, Amsterdam, 1980.

E.T. Jaynes. *Papers on Probability, Statistics and Statistical Physics (ed. by R.D. Rosencrantz)*. Reidel, Dordrecht, 1983.

E.T. Jaynes. Probability Theory as Logic. In *Proceedings of the Ninth Annual Workshop on Maximum Entropy and Bayesian Methods*, Dordrecht, Holland, 1990. Kluwer Academic Press. present version substantially revised, corrected, and extended 5/1/94.

E.T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, 2003.

A.H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, London, 1970.

H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London (Series A)*, 186:453–461, 1946.

H. Jeffreys. *Theory of Probability*. Oxford University Press, London, 1939.

H. Jeffreys. *Theory of Probability (Third Edition)*. Oxford University Press, London, 1961.

Adam M. Johansen and Arnaud Doucet. Auxiliary variable sequential Monte Carlo methods. Technical report, University of Bristol, Statistics Group, 2007.

Tor A. Johansen and Bjarne A. Foss. ORBIT — Operating-Regime-Based Modeling and Identification Toolkit, 1998. Preprint submitted to Elsevier Preprint 18 July 1998.

Tor A. Johansen and Erik Weyer. On Convergence Proofs in System Identification — A General Principle using ideas from Learning Theory, 1997. Preprint submitted to Elsevier Preprint 25 November 1997.

Tor A. Johansen, Robert Shorten, and Roderick Murray-Smith. On the interpretation and identification of dynamic Takagi-Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*, 8(3):297–313, 2000.

Tor Arne Johansen and Roderick Murray-Smith. The operating regime approach to nonlinear modelling and control. In *Roderick Murray-Smith and Tor Arne Johansen (eds.)*, Multiple Model Approaches to Modelling and Control, pages 3–72. Taylor and Francis, London, 1997.

L. Jones. A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *The Annals of Statistics*, 20 (1):608–613, 1992.

L.K. Jones. Local Greedy Approximation for Nonlinear Regression and Neural Network Training. *The Annals of Statistics*, 28(5):1379–1389, 2000.

M.C. Jones. Randomly choosing parameters from the stationarity and invertibility region of autoregressive-moving average models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 36(2):134–138, 1987.

Anatoli Juditsky, Håkan Hjalmarsson, Albert Benveniste, Bernard Delyon, Lennart Ljung, Jonas Sjöberg, and Qinghua Zhang. Nonlinear black-box models in system identification: Mathematical foundations. *Automatica*, Special issue on Trends in System Identification, 31(12):1725–1750, 1995.

R.E. Kass and E. Raftery. Bayes factors. *Journal of the American Statistical Society*, 90: 773–795, 1995.

Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819, 2005.

Clemens Kirchmair. *Identifikation von Systemen mit Hysterese mit Hilfe von Preisach-Neuronen in vorstrukturierten neuronalen Netzen*, volume 258 of *Dissertationen zur künstlichen Intelligenz*. Akademische Verlagsgesellschaft Aka GmbH, Berlin, 2002.

G. Kitagawa. Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82:1032–1063, 1987.

A. Kong, J.S. Liu, and W.H. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89:278–288, 1994.

S.V. Konyagin and V.N. Temlyakov. Rate of convergence of Pure Greedy Algorithm. *East Journal on Approximations*, 5:493–499, 1999.

Jelena Kovačević and Wim Sweldens. Wavelet Families of Increasing Order in Arbitrary Dimensions. *IEEE Transactions on Image Processing*, 9(3):480–496, 2000. doi: 10.1109/83.826784.

M.A. Krasnosel'skiĭ and A.V. Pokrovskiĭ, editors. *Systems with hysteresis*. Springer-Verlag, Berlin, 1989.

Pavel Krejčí. Evolution variational inequalities and multidimensional hysteresis operators. In *Pavel Drábek, Pavel Krejčí and Peter Takáč (eds.),* Nonlinear Differential Equations, volume 404 of *CRC Research Notes in Mathematics*, pages 47–110. Chapman & Hall, Boca Raton, 1999.

Hans R. Künsch. Recursive Monte Carlo Filters: Algorithms and Theoretical Analysis. *The Annals of Statistics*, 33(5):1983–2021, 2005. doi: 10.1214/009053605000000426.

Hans R. Künsch. State Space and Hidden Markov Models. In *Ole E. Barndorff-Nielsen, David R. Cox and Claudia Klüppelberg (eds.),* Complex Stochastic Systems, pages 109–173. Chapman & Hall/CRC, Boca Raton, 2001.

G. Kyriazis. Wavelet coefficients measuring smoothness in $H_p(\mathbb{R}^d)$. *Applied and Computational Harmonic Analysis*, 3:100–119, 1996.

Steffen L. Lauritzen. Causal Inference from Graphical Models. In *Ole E. Barndorff-Nielsen, David R. Cox and Claudia Klüppelberg (eds.)*, Complex Stochastic Systems, pages 63–107. Chapman & Hall/CRC, Boca Raton, 2001.

P. Lindskog and L. Ljung. Tools for Semiphysical Modelling. *International Journal of Adaptive Control and Signal Processing*, 9(6):509–523, 1995.

Jane Liu and Mike West. Combined Parameter and State Estimation in Simulation-Based Filtering. In *Arnaud Doucet, Nando de Freitas and Neil Gordon (eds.)*, Sequential Monte Carlo Methods in Practice, pages 197–223. Springer Verlag, New York, 2001.

J.S. Liu. Metropolized independent sampling with comparison to rejection sampling and importance sampling. *Statistics and Computing*, 6:113–119, 1996.

J.S. Liu and R. Chen. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576, 1995.

J.S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.

E.D. Livshitz and V.N. Temlyakov. Two lower estimates in greedy approximation. *Constructive Approximation*, 19(4):509–524, 2003.

Lennart Ljung. *System Identification: Theory for the User (Second Edition)*. Prentice Hall PTR, Upper Saddle River, New Jersey, 1999.

Lennart Ljung and Torsten Söderström, editors. *Theory and Practice of Recursive Identification*. The MIT Press, Cambridge / London, 1983.

Marco J. Lombardi and Simon J. Godsill. On-line Bayesian estimation of AR signals in symmetric alpha-stable noise. Working Paper 2004/05, Università degli Studi di Firenze, 2004.

Alfred Karl Louis, Peter Maaß, and Andreas Rieder, editors. *Wavelets: Theorie und Anwendungen*. Teubner, Stuttgart, 1994.

F. Low. Complete sets of wave packets. In *C. DeTar (ed.)*, A Passion for Physics — Essays in Honor of Geoffrey Chew, pages 17–22. World Scientific, Singapore, 1985.

Ronald Mahler. Random Sets: Unification and Computation for Information Fusion — A Retrospective Assessment. In *Proceedings of the Seventh International Conference on Information Fusion, International Society of Information Fusion. Mountain View, CA.*, pages 1–20, 2004.

S. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L_2(\mathbb{R})$. *Transactions of the American Mathematical Society*, 315:69–87, 1989.

S. Mallat. *A Wavelet Tour of Signal Processing, Second Edition*. Academic Press, San Diego, 1999.

I.D. Mayergoyz, editor. *Mathematical models of hysteresis*. Springer Verlag, New York, 1991.

I.D. Mayergoyz, editor. *Mathematical models of hysteresis and their applications. Second edition*. Elsevier Science Inc., New York, 2003.

S.N. McEachern, M. Clyde, and J.S. Liu. Sequential importance sampling for nonparametric Bayes models: The next generation. *Canadian Journal of Statistics*, 27(2):251–267, 1999.

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

S.P. Meyn and R.L. Tweedie, editors. *Markov chains and stochastic stability*. Springer Verlag, New York, 1993.

S. Mollov, P. van der Veen, R. Babuška, J. Abonyi, J.A. Roubos, and H.B. Verbruggen. Extraction of Local Linear Models from Takagi-Sugeno Fuzzy Model with Application to Model-based Predictive Control. In *7th European Conference on Intelligent Techniques and Soft Computing (EUFIT '99), Aachen, Germany*, 1999.

J.F. Monahan. A note on enforcing stationarity in autoregressive-moving average models. *Biometrika*, 71(2):403–404, 1984.

Pierre Del Moral. *Feynman-Kac Formulae*. Probability and its Applications. Springer-Verlag, Berlin Heidelberg New York, 2004.

P. Moulin and J. Liu. Analysis of multiresolution image denoising schemes using generalized Gaussian and complexity priors. *IEEE Transactions on Information Theory*, 45(3):909–919, 1999.

P. Müller. Posterior integration in dynamic models. *Computing Science and Statistics*, 24: 318–324, 1992.

Peter Müller and Brani Vidakovic, editors. *Bayesian Inference in Wavelet-Based Models*, volume 141 of *Lecture Notes in Statistics*. Springer-Verlag, New York, 1999.

David Mumford. Pattern Theory: the Mathematics of Perception. In *ICM*, volume III, pages 1–3, 2002.

David Mumford and Agnes Desolneux. Pattern Theory through Examples, in preparation.

Roderick Murray-Smith and Tor Arne Johansen, editors. *Multiple Model Approaches to Modelling and Control*. Taylor and Francis, London, 1997.

Sreerama K. Murthy, Simon Kasif, and Steven Salzberg. A System for Induction of Oblique Decision Trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.

G.P. Nason. Choice of the threshold parameters in wavelet function estimation. In *A. Antoniadis and G. Oppenheim (eds.)*, Wavelets and Statistics, pages 261–280. Springer-Verlag, New York, 1995.

G.P. Nason. Wavelet shrinkage using cross-validation. *Journal of the Royal Statistical Society, Series B*, 58(2):463–479, 1996.

Oliver Nelles. LOLIMOT — Lokale, lineare Modelle zur Identifikation nichtlinearer, dynamischer Systeme. *Automatisierungstechnik*, 45(4):163–174, 1997.

Oliver Nelles. *Nonlinear System Identification*. Springer-Verlag, Berlin, 2001.

Oliver Nelles, Alexander Fink, and Rolf Isermann. Local Linear Model Trees (LOLIMOT) Toolbox for Nonlinear System Identification. Technical report, Institute of Automatic Control (IFAC), Darmstadt University of Technology, 2000.

O. Nerrand, P. Roussel-Ragot, L. Personnaz, G. Dreyfus, and S. Marcos. Neural networks and non-linear adaptive filtering: Unifying concepts and new algorithms. *Neural Computation*, 5(99):165–197, 1993.

Helmut Neunzert and Bernd Rosenberger. *Schlüssel zur Mathematik*. ECON Verlag, Düsseldorf, 1991.

Helmut Neunzert and Bernd Rosenberger. *Stichwort Mathematik*. Paperback edition of Neunzert and Rosenberger [1991]. Droemersche Verlagsanstalt, München, 1993.

E. Nummelin. *General Irreducible Markov Chains and Non-Negative Operators*. Cambridge University Press, Cambridge, 1984.

T. Ogden and E. Parzen. Change-point approach to data analytic wavelet thresholding. *Statistics and Computing*, 6(2):93–99, 1996a.

T. Ogden and E. Parzen. Data dependent wavelet thresholding in nonparametric regression with change-point applications. *Computational Statistics and Data Analysis*, 22:53–70, 1996b.

Jimmy Olsson, Olivier Cappé, Randal Douc, and Éric Moulines. Sequential Monte Carlo Smoothing with application to parameter estimation in non-linear state space models. Technical report, Lund University, 2006.

K. Oskolkov. Polygonal approximation of functions of two variables. *Mathematics of the USSR – Sbornik*, 35:851–861, 1979.

J.B. Paris. *The Uncertain Reasoner's Companion*. Cambridge University Press, Cambridge, 1994.

J. Peetre. A Theory of Interpolation of Normed Spaces. Course Notes, University of Brasilia, 1963.

P.H. Peskun. Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, 60(3):607–612, 1973.

P.P. Petrushev. Direct and converse theorems for spline and rational approximation and Besov spaces. In *M. Cwikel et al. (eds.),* Function Spaces and Applications: Proceedings of the US-Swedish Seminar held in Lund, Sweden, 1986, *Vol. 1302 of Lecture Notes in Mathematics*, pages 363–377. Springer, Berlin, 1988.

G. Pisier. Remarques sur un résultat non publié de B. Maurey. Seminaire d'Analyse Fonctionelle 1980-81, École Polytechnique, Centre de Mathématiques, Palaiseau, 1980.

M.K. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, 94:590–599, 1999.

L.R. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

H. Raiffa and R. Schlaifer. *Applied Statistical Decision Theory*. Division of Research, Graduate School of Business Administration, Harvard University, Boston, 1961.

B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, 1996.

H. Robbins. Asymptotically subminimax solutions of compound statistical decision problems. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability (Jerzy Neyman ed.), University of California Press, Berkeley and Los Angeles*, pages 131–148, 1951.

H. Robbins. An empirical Bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1 (Jerzy Neyman ed.), University of California Press, Berkeley and Los Angeles*, pages 157–163, 1956.

Christian P. Robert. *The Bayesian Choice (Second Edition)*. Springer-Verlag, New York, 2001.

Gareth O. Roberts and Jeffrey S. Rosenthal. General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1:20–71, 2004. doi: 10.1214/154957804100000024.

J.S. Rosenthal. Minorization conditions and convergence rates for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 90:558–566, 1995.

J.S. Rosenthal. A review of asymptotic convergence for general state space Markov chains. *Far East Journal of Theoretical Statistics*, 5:37–50, 2001.

D.B. Rubin. A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: the SIR algorithm (comment on M.A. Tanner and W.H. Wong: The calculation of posterior distributions by data augmentation). *Journal of the American Statistical Association*, 82(398):543–546, 1987.

E. Schmidt. Zur Theorie der lineraren und nichtlinearen Integralgleichungen. I. Teil. *Mathematische Annalen*, 63(4):433–476, 1907.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

Clayton Scott and Robert Nowak. Minimax-Optimal Classification with Dyadic Decision Trees. *IEEE Transactions on Information Theory*, 52(4):1335–1353, 2006. doi: 10.1109/TIT.2006.871056.

C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27: 379–423 and 623–656, 1948.

N. Shephard. Statistical aspects of ARCH and stochastic volatility. In *D.R. Cox, D.V. Hinkley and O.E. Barndorff-Nielsen (eds.)*, Time Series Models with Econometric, Finance and Other Fields, pages 1–67. Chapman and Hall, London, 1996.

J. Sjöberg, H. Hjalmarsson, and L. Ljung. Neural networks in system identification. Technical Report Nr. LiTH-ISY-R-1622, Linköping University, 1994. `ftp://ftp.control.isy.liu.se/pub/Reports/1994/1622.ps.Z`.

Jonas Sjöberg. A Nonlinear Grey-Box Example Using a Stepwise System Identification Approach. In *Proceedings of the 11th IFAC Symposium on Identification, Santa Barbara, USA*, 2000.

Jonas Sjöberg, Qinghua Zhang, Lennart Ljung, Albert Benveniste, Bernard Delyon, Pierre-Yves Glorennec, Håkan Hjalmarsson, and Anatoli Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, Special issue on Trends in System Identification, 31(12):1691–1724, 1995.

Paul Snow. On the correctness and reasonableness of Cox's theorem for finite domains. *Computational Intelligence*, 14(3):452–459, 1998.

James C. Spall, editor. *Bayesian Analysis of Time Series and Dynamic Models*, volume 94 of *STATISTICS: Textbooks and Monographs*. Marcel Dekker, Inc., New York Basel, 1988.

C. Stein. Estimation of the mean of a multivariate normal distribution. *Annals of Statistics*, 9 (6):1135–1151, 1981.

G. Strang and G. Fix. A Fourier analysis of the finite-element variational method. In *G. Geymonat (ed.)*, Constructive Aspects of Functional Analysis, pages 795–840. C.I.M.E., Edizioni Cremonese, Rome, 1973.

Wim Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, 1997.

Wim Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Journal of Applied and Computational Harmonic Analysis*, 3(2):186–200, 1996.

V. Temlyakov. The best *m*-term approximation and greedy algorithms. *Advances in Computational Mathematics*, 8(3):249–265, 1998.

V.N. Temlyakov. Nonlinear Methods of Approximation. *Foundations of Computational Mathematics*, 3:33–107, 2002.

Elizabeth A. Thompson. Monte Carlo Methods on Genetic Structures. In *Ole E. Barndorff-Nielsen, David R. Cox and Claudia Klüppelberg (eds.),* Complex Stochastic Systems, pages 175–218. Chapman & Hall/CRC, Boca Raton, 2001.

Herbert J.A.F. Tulleken. Grey-box modelling and identification using physical knowledge and bayesian techniques. *Automatica*, 29(2):285–308, 1993.

Jürgen van Gorp and Johan Shoukens. A scheme for nonlinear modeling. In *World Multiconference on Systemics, Cybernetics and Informatics (SCI '99), 5th International Conference on Information Systems Analysis and Synthesis (ISAS '99), Orlando, USA*, volume 5, pages 450–456, July–August 1999.

B. Vidakovic. Nonlinear wavelet shrinkage with Bayes rules and Bayes factors. *Journal of the American Statistical Association*, 93(441):173–179, 1998.

Augusto Visintin. *Differential Models of Hysteresis*, volume 111 of *Applied Mathematical Sciences*. Springer-Verlag, Berlin Heidelberg, 1994.

Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

Peter Walley. Measures of uncertainty in expert systems. *Artificial Intelligence*, 83(1):1–58, 1996.

Guido Walz, editor. *Lexikon der Mathematik,* volume 1–6. Spektrum Akademischer Verlag, Heidelberg, 2000-2003.

P. Weiss and J. de Freudenreich. Initial magnetization as a function of the temperature. *Archives des Sciences Physiques et Naturelles*, 42:449–470, 1916.

Jochen Werner. *Numerische Mathematik. Band 1*. Vieweg, Braunschweig / Wiesbaden, 1992a.

Jochen Werner. *Numerische Mathematik. Band 2*. Vieweg, Braunschweig / Wiesbaden, 1992b.

M. West. Mixture models, Monte Carlo, Bayesian updating and dynamic models. *Computer Science and Statistics*, 24:325–333, 1993.

Mike West and Jeff Harrison, editors. *Bayesian Forecasting and Dynamic Models*. Springer Series in Statistics. Springer-Verlag, New York, second edition, 1997.

N. Weyrich and G.T. Warhola. De-noising using wavelets and cross-validation. In *S.P. Singh (ed.),* Approximation Theory, Wavelets and Applications, pages 523–532. Kluwer, Dordrecht, 1995.

P. Whittle. *Optimal Control: Basics and Beyond*. Wiley, Chichester, UK, 1996.

J. Wloka. *Partielle Differentialgleichungen*. Teubner, Leipzig, 1982.

V.S. Zaritskii, V.B. Svetnik, and L.I. Shimelevich. Monte Carlo technique in problems of optimal data processing. *Automation and Remote Control*, 12:95–103, 1975.

Index of definitions

α-Hölder continuous, ***320***
θ, q-norm, ***241***
φ-irreducibility, ***161***
φ-irreducible, ***163***
0-1 loss, ***132***

absolute error loss, ***132***
accept-reject algorithm, ***176***
acceptance-rejection method, ***176***
accessible sets, ***164***
actions, ***127***
adaptive pursuit, ***259***
admissibility condition, ***218***
admissible, ***130***
Akaike Information Criterion, AIC, ***24***
alternating sequences, ***89***
analysis filter bank, ***226***
antisymmetric extension, ***100***
aperiodic, ***167***
approximants, ***237***
approximation error, ***237***
approximation error (for libraries), ***256***
approximation space, ***238***
Armijo step size, ***54***
associated family of wavelet coefficients, ***287***
associated family of weight functions, ***35***
associated half space, ***60***
associated hyperplane, ***59***
associated polytope, ***60***
associated string, ***34***
assumptions, ***3***, ***122***
atomic decomposition, ***73***
atoms, ***73***
automaton, ***68***
autoregressive model, ***153***
autoregressive moving average model, ***157***
Auxiliary Particle Filter (APF), ***203***
average loss, ***128***

Bayes risk, ***128, 129***
Bayes rule, ***129***
Bayes' theorem, ***125***
Bayesian Decision Theory, ***127***
Bayesian Information Criterion, BIC, ***25***
Bayesian principle, ***128***
Bayesian statistical model, ***125***
Bernstein inequality, ***242***
Besov norm, ***245***
Besov space, ***245***
binary, ***34***
biorthogonal, ***323***
biorthogonal wavelets, ***221***
black-box model, ***3***
bootstrap filter, ***201***
bounds of the frame, ***220***

canonical form, ***14***
canonical ordering, ***33***
Causality axiom, ***67***
chain components, ***170***
chain graph, ***170***
child, ***31***
children, ***33***
clique, ***31***
closedness assumption, ***138***
Cocycle property, ***67***
complete, ***68***
complete graph, ***30***
complete memory sequence, ***86***
components, ***36***
computation, ***119***
conditioning by intervention, ***123***

conditioning by observation, ***123***
congruency property, ***92***
conjugate for a likelihood function, ***141***
connected, ***32***
connected component, ***32***
Consistency axiom, ***67***
continuous piecewise monotone functions, ***77***
continuous wavelet transform (CWT), ***218***
convex (open) polytope, ***59***
corner, ***94***
countable, ***158***
cumulant generating function, ***142***
cycle, ***32***
cyclic kernel, ***178***

d-cycle, ***167***
data, ***122***
decision, ***127***
decision map, ***35***
decision procedure, ***127***
decision rule, ***127***
decision theory, ***123***
decision tree, ***35***
decision tree based weight functions, ***38***
decomposition filter bank, ***226***
decomposition principle, ***72***
decreasing rearrangement, ***243***
degree, ***32***
delayed relay, ***78***
density, ***320***
descent direction, ***52***
detailed balance, ***176***
deterministic state space system, ***65***
dictionary, ***257***
difference dynamics, ***8***
difference operator, ***245***
differential dynamical system, ***69***
diffusion process, ***185***
dilate, ***222***
dilation matrix, ***222***
direct theorem, ***242***
directed acyclic graph (DAG), ***32***
directed edge, ***31***
directed graph, ***31***
Discrete Wavelet Transform (DWT), ***220***
dual basis, ***323***
dyadic partitioning, ***20***
dyadic wavelet frames, ***221***
dynamic model, ***153***
dynamical system, ***7***, ***65***

edge, ***30***
edge weight, ***35***
Effective Sample Size (ESS), ***198***
empirical Bayes, ***149***
endvertex, ***30***
entropy, ***140***
entropy distance, ***132***
enumeration, ***34***
equivariant decision rules, ***145***
ergodicity, ***162***
error function, ***7***, ***19***
error of *n*-term approximation, ***238***
esimate, ***127***
estimation, ***127***
estimator, ***127***
evanescence, ***162***
exchangeable, ***139***
expectation maximization (EM) algorithm, ***192***
exponential family, ***141***
Extended Kalman Filter (EKF), ***194***
extended probability model, ***139***

Factorization, ***169***
Fast Wavelet Transform (FWT), ***226***
filtering, ***186***
filtering density, ***188***
first hitting time, ***161***
first return, ***161***
first stage weight, ***203***
Fixed-interval smoothing, ***188***
Fixed-lag smoothing, ***188***
forced motion, ***72***
forest, ***32***
Fourier transform, ***324***
Fourier transformed function, ***324***

Fourier transformed measure, ***325***
free motion, ***72***
frequentist principle, ***128***
frequentist risk, ***128***
full, ***34***
full binary decision trees, ***35***
full conditional distribution, ***148***, ***168***, ***178***

Gabor transform, ***213***
Gauß-Newton descent direction, ***53***
general, ***158***
general detatiled balance condition, ***180***
general state space model, ***183***
Generalized Bayes Formula, ***126***
Generalized Bayes Rule, ***129***
generalized local model network, ***74***
generates, ***220***
generation of observations, ***184***
Gibbs sampler, ***176***
global basis functions, ***12***
Global Markov property, ***169***
gradient descent direction, ***53***
graph, ***30***
graphical model, ***168***
grey-box model, ***4***

Haar basis, ***221***
Haar function, ***221***
half space, ***59***
hard and soft thresholding estimators, ***265***
hard thresholding operator, ***253***
Harris (recurrent), ***166***
Harris recurrent, ***166***
has a value, ***130***
Hausdorff-Young inequality, ***325***
height, ***34***
Heisenberg uncertainty principle, ***213***
Hellinger distance, ***132***
hidden, ***8***, ***60***
hidden layer, ***13***
Hidden Markov Model (HMM), ***184***
Hierarchical Bayes model, ***148***
high pass filter coefficients, ***225***
highest probability density (HPD) region, ***134***
highly nonlinear approximation, ***236***
highly nonlinear problem, ***255***
history, ***196***
hyperparameters, ***148***
hyperplane, ***59***
hysteron, ***78***
hysterons, ***73***
HySyWaT, ***292***

identification error, ***45***
identification set, ***45***
imperfect knowledge, ***120***
implementation, ***119***
Importance Sampling (IS), ***172***
improper prior distribution, ***126***
inadmissible, ***130***
incremental weight, ***196***
indegree, ***31***
independence Metropolis-Hastings, ***178***
induced, ***31***
infinitesimal, ***139***
information independent, ***138***
initial distribution, ***159***
initial state, ***8***, ***66***
initial time, ***66***
inner dynamics, ***8***
inner hypercuboids, ***290***
inner node, ***33***
inner vertex, ***33***
input function, ***66***
input function space, ***65***
input value space, ***65***
input-output operator, ***68***
interpolation spaces, ***240***
interpretation, ***119***
interval, ***66***
Interval axiom, ***66***
intrinsic credible region, ***134***
intrinsic loss, ***132***
invariant, ***145***, ***165***
invariant under $\mathscr{G}$, ***144***
invariant under orthogonal transformations, ***268***

invariant under the action of the group $\tilde{\mathscr{G}}$, ***145***
invariant under the action of the group $\mathscr{G}$, ***144***
invariant with respect to the time transformation, ***70***
Inverse Discrete Wavelet Transform (IDWT), ***220***
inverse Fourier transform, ***324***
inverse partial autocorrelations, ***157***
inverse theorem, ***242***
irreducibility, ***163***
irreducibility measure, ***163***

Jackson inequality, ***242***
Jeffreys' noninformative prior, ***146***
joint admissibility condition, ***219***
joint distribution, ***125***
joint smoothing density, ***187***

k-ary, ***34***
K-functional, ***240***
$\mathbb{K}$-linear, ***71***
k-regular, ***34***
Kalman filter, ***193***
Kalman gain matrix, ***193***
Kalman smoother, ***193***
knowledge, ***122***
Koksma-Hlawka inequality, ***175***
Kullback-Leibler divergence, ***132***

Laplace's prior, ***144***
leaf, ***33***
left child, ***34***
lemma of Riemann-Lebesgue, ***325***
level, ***34***
Levenberg-Marquardt descent direction, ***53***
library, ***255***
life span, ***66***
likelihood, ***124***, ***192***
Likelihood Principle, ***136***
linear, ***10***
linear approximation, ***237***
linear combination of basis functions, ***11***
linear difference dynamical system, ***73***
linear differential dynamical system, ***72***
linear model, ***185***
linear system, ***4***
linear time-invariant, ***73***
linearity, ***4***
link, ***30***
Lipschitz boundary, ***321***
Lipschitz domain, ***246***, ***321***
Lipschitz-continuous, ***320***
local basis functions, ***12***
Local Markov property, ***169***
local memory, ***74***
local model network (LMN), ***14***
localized at the frequency, ***213***
localized at the phase point, ***213***
localized at the time, ***213***
location parameter, ***11***, ***17***, ***145***
logic, ***122***
logistic function, ***39***
LOLIMOT algorithm, ***19***
long-time memory, ***74***
loop, ***31***
Lorentz space, ***243***
loss function, ***127***
low pass filter coefficients, ***225***
lower threshold, ***78***
lowest expected loss (LEL) region, ***134***
LTI, ***73***

marginal distribution, ***125***
marginalization, ***125***
Markov Chain Monte Carlo (MCMC) methods, ***176***
Markov transition function, ***159***
matching prior, ***146***
matching pursuit, ***259***
mathematical model, ***119***
maximal irreducibility measure, ***164***
maximin risk, ***130***
maximum a posteriori (MAP) estimator, ***133***
maximum entropy prior, ***140***
maximum likelihood (ML) estimator, ***192***
Mayergoyz representation theorem, ***93***
mean of wavelet trees, ***290***

measurable space, ***319***
measure space, ***319***
measurement errors, ***119***
measurement noise, ***119***
Metropolis method, ***177***
Metropolis-Hastings sampler, ***177***
minimax risk, ***129***
minimax rule, ***129***
mixture kernel, ***178***
model, ***3***
model errors, ***119***
model selection, ***24***
modelling, ***119***
modulus of smoothness, ***245***
Monotonicity assumption, ***138***
monotonicity partition, ***77***
Monte Carlo (MC) methods, ***171***
mother basis function, ***11***
mother wavelet, ***218***
move types, ***181***
moving average model, ***155***
μ-continuous, ***319***
μ-singular, ***319***
multilayer network, ***13***
multinomial sampling, ***174***
multiresolution analysis (MRA), ***222***

n-step transition probability kernel, ***160***
n-term approximation, ***235***
NARX, ***8***
natural, ***141***
natural parameter space, ***142***
near best approximation, ***242***
negative saturation, ***85***
neighbours, ***31***
neurons, ***15***
Newton descent direction, ***53***
node, ***30***
NOE, ***9***
non-parametric regression problem, ***263***
non-randomized, ***127***
non-tangential maximal function, ***322***
nonlinear approximation, ***237***
nonlinear equation error model, ***8***
nonlinear system, ***5***
nonlocal memory, ***74***
nonnegative garrote, ***265***
normal (or Gaussian) linear model, ***185***
normalized, ***15***, ***17***, ***35***
normalized importance weights, ***173***
null, ***165***

observations, ***124***
observed data, ***3***
occupation time, ***161***
order, ***217***
orientation, ***34***
orthogonal greedy algorithm (OGA), ***261***
outdegree, ***31***
outer dynamics, ***8***
outer hypercuboids, ***290***
outliers, ***185***
output, ***66***
output function, ***8***
output map, ***65***
output value space, ***65***

Pairwise Markov property, ***169***
parametric empirical Bayes, ***150***
parametric statistical model, ***124***
parent, ***31***
parents, ***33***
partial autocorrelations, ***154***
particle filter, ***197***
particle paths, ***196***
particles, ***196***
path, ***32***, ***196***
path space, ***158***
period, ***167***
periodic wavelets, ***231***
phase space, ***212***
phase space representation, ***212***
piecewise monotone function, ***77***
population discrete wavelet coefficients, ***264***
positive, ***165***
positive Harris, ***166***
positive recurrent, ***165***
positive saturation, ***85***

posterior distribution, ***125***
posterior expected loss, ***128***
posterior median, ***272***
posterior odds ratio, ***272***
prediction, ***186***
predictive distribution, ***126***
Preisach half plane, ***79***
Preisach model, ***80***
Preisach operator, ***81***
Preisach weight, ***80***
primitive function, ***93***
primitive functions, ***94***
principal shift invariant (PSI) space, ***222***
prior distribution, ***125***
prior knowledge, ***3***
probability centred *q*-credible region, ***134***
probability tree, ***35***
projection pursuit, ***259***
proper, ***34***, ***126***
proposal, ***177***
proposal density, ***177***
proposal distribution, ***172***
pure greedy algorithm (PGA), ***259***

q-credible region, ***134***
quadratic loss, ***131***

radial basis functions, ***17***
radial construction, ***13***
radial maximal function, ***322***
Radon-Nikodym derivative, ***320***
random walk Metropolis, ***179***
random walk on the log-scale, ***179***
randomized decision rule, ***127***
range invariant, ***71***
Rao-Blackwellisation, ***204***
rate independence, ***92***
real system, ***3***, ***118***
realizations, ***158***
reconstruction filter bank, ***226***
recurrence, ***162***
recurrent, ***164***
recurrent neural networks, ***14***
reduced memory sequences, ***87***
redundancy, ***258***
reference prior, ***146***
Refinability assumption, ***138***
reflection coefficients, ***154***
regime based weight functions, ***15***
regime vector, ***17***
regression vector, ***16***
regressor matrix, ***152***
regular, ***222***
(regular) grid, ***222***
regularization parameter, ***54***
reiteration theorem, ***241***
rejection sampling, ***176***
relaxation parameter, ***261***
relaxed greedy algorithm (RGA), ***260***
representation theorem, ***139***
resampling, ***173***
residual sampling, ***175***
reversible, ***68***
ridge construction, ***13***
ridge function, ***13***
Riesz Basis, ***323***
right child, ***34***
Risk Inflation Criterion, RIC, ***25***
root, ***33***
rooted tree, ***33***

sample density, ***124***
sample discrete wavelet coefficients, ***264***
sample paths, ***158***
sample ties, ***199***
Sampling/Importance Resampling (SIR), ***173***
scale invariant, ***144***
scale parameter, ***11***, ***17***, ***145***
scaling function, ***222***
Schwarz's Information Criterion, SIC, ***25***
selection, ***173***
separability assumption, ***138***
separable wavelets, ***229***
sequence space, ***158***
Sequential Importance Sampling (SIS), ***196***
Sequential Importance Sampling with Replacement (SISR), ***197***
Sequential Monte Carlo (SMC), ***195***

Sequential Monte Carlo Samplers, ***195***
shift, ***222***
shift (translation), ***326***
short-time memory, ***74***
shrinkage rule, ***265***
sigmoid function, ***12***
signal, ***211***
signed measures, ***319***
simple graph, ***31***
simulated outputs, ***9***
slice sampler, ***182***
smoothing, ***186***
smoothing density, ***188***
Sobolev embedding theorem, ***246***
Sobolev space, ***244***
soft operator decision trees (SODT), ***39***
space of functions of bounded mean oscillation, *BMO*, ***322***
space of piecewise monotone functions, ***77***
sparse, ***264***
sparse coefficient spaces, ***288***
sparse spaces, ***288***
splitting rule, ***38***
stable on a space, ***242***
state, ***8***, ***66***
state domain, ***8***
state evolution, ***184***
state of nature, ***127***
state space, ***65***, ***158***
state space model, ***8***
state space system, ***8***
state trajectory, ***66***
state transition function, ***8***
state transition map, ***65***
state transition operator, ***8***
static system, ***6***
stationary, ***153***
step size, ***52***
stochastic volatility model, ***185***
stopping times, ***160***
Strang-Fix condition, ***223***
stratified sampling, ***175***
strict, ***34***
subgraph, ***31***
superposition principle, ***72***
symmetric, ***144***
synthesis filter bank, ***226***
systematic sampling, ***175***

tangent filter, ***206***
tapped delays, ***8***
target function, ***234***, ***237***
tensor product construction, ***12***
test functions, ***212***
test set, ***45***
theorem of Plancherel, ***325***
thresholding, ***253***
thresholding rule, ***265***
tight, ***220***
tightness, ***162***
time domain, ***65***
time invariant, ***71***
time scaling, ***70***
time series model, ***153***
time shift, ***70***
time transformation, ***70***
(time)-continuous models, ***7***
(time)-discrete models, ***7***
time-homogeneous Markov chain, ***159***
time-reversibility, ***176***
topological, ***158***
total variation norm, ***166***
trail, ***32***
trajectory, ***196***
transient, ***164***
transition probability kernel, ***159***
translation invariant, ***144***, ***326***
translation operator, ***245***
trapezoid, ***94***
tree, ***32***
triangle, ***94***

undirected edge, ***31***
undirected graph, ***31***
uniformly transient, ***164***
universal threshold, ***266***
unnormalized importance weights, ***172***
upper threshold, ***78***

validation error, ***46***
validation set, ***45***
vanishing moments, ***217***
vertex, ***30***

wavelet, ***218***
wavelet basis, ***224***
wavelet coefficients, ***224***
wavelet frame, ***220***
wavelet packet bases, ***232***
wavelet subtree, ***287***
wavelet transformed function, ***218***
wavelet tree, ***287***
weak L^p, ***243***
weight degeneracy, ***197***
weighted empirical distribution, ***172***
white noise, ***184***
white-box model, ***3***
window, ***214***
windowed Fourier transform, ***213***
wiping-out property, ***92***
Wold decomposition, ***155***

Wissenschaftlicher Werdegang

- 1992 Abitur am Realgymnasium in Lebach/Saar
- 1993–2002 Studium im Diplomstudiengang Mathematik mit Anwendungsfach Informatik an der Universität Kaiserslautern (jetzt Technische Universität Kaiserslautern)
- 2002 Diplom
- 2003–2006 Stipendiat des Graduiertenkollegs „Mathematik und Praxis“ des Fachbereichs Mathematik an der TU Kaiserslautern
- seit 2007 Wissenschaftlicher Mitarbeiter am Fraunhofer Institut für Techno- und Wirtschaftsmathematik (ITWM) in Kaiserslautern

Scientific Career

- 1992 Abitur at “Realgymnasium” in Lebach/Saar (Germany)
- 1993–2002 Studies in Mathematics with minor subject Computer Science at the University of Kaiserslautern (Germany), now Technical University of Kaiserslautern
- 2002 German Diplom
- 2003–2006 Scholarship of the Graduate Research Training Programme “Mathematics and Practice” at the Department of Mathematics at the TU Kaiserslautern
- since 2007 Scientific employee with the “Fraunhofer Institut für Techno- und Wirtschaftsmathematik” (ITWM; Fraunhofer Institute for Industrial Mathematics) at Kaiserslautern (Germany)