Linköping studies in science and technology. Dissertations.
No. 1351

Regularization for Sparseness and Smoothness

Applications in System Identification and Signal Processing

Henrik Ohlsson

Department of Electrical Engineering
Linköping University, SE–581 83 Linköping, Sweden

Linköping 2010

Linköping studies in science and technology. Dissertations.
No. 1351

Regularization for Sparseness and Smoothness – Applications in System Identification and Signal Processing

Henrik Ohlsson

ohlsson@isy.liu.se
www.control.isy.liu.se
Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE–581 83 Linköping
Sweden

ISBN 978-91-7393-287-5 ISSN 0345-7524

Printed by LiU-Tryck, Linköping, Sweden 2010

To family and friends!

Abstract

In system identification, the *Akaike Information Criterion* (AIC) is a well known method to balance the model fit against model complexity. Regularization here acts as a price on model complexity. In statistics and machine learning, regularization has gained popularity due to modeling methods such as *Support Vector Machines* (SVM), *ridge regression* and *lasso*. But also when using a *Bayesian* approach to modeling, regularization often implicitly shows up and can be associated with the prior knowledge. Regularization has also had a great impact on many applications, and very much so in clinical imaging. In *e.g.,* breast cancer imaging, the number of sensors is physically restricted which leads to long scan times. Regularization and sparsity can be used to reduce that. In *Magnetic Resonance Imaging* (MRI), the number of scans is physically limited and to obtain high resolution images, regularization plays an important role.

Regularization shows-up in a variety of different situations and is a well known technique to handle ill-posed problems and to control for overfit. We focus on the use of regularization to obtain sparseness and smoothness and discuss novel developments relevant to system identification and signal processing.

In regularization for sparsity a quantity is forced to contain elements equal to zero, or to be sparse. The quantity could *e.g.,* be the regression parameter vector of a linear regression model and regularization would then result in a tool for variable selection. Sparsity has had a huge impact on neighboring disciplines, such as machine learning and signal processing, but rather limited effect on system identification. One of the major contributions of this thesis is therefore the new developments in system identification using sparsity. In particular, a novel method for the estimation of segmented ARX models using regularization for sparsity is presented. A technique for piecewise-affine system identification is also elaborated on as well as several novel applications in signal processing. Another property that regularization can be used to impose is smoothness. To require the relation between regressors and predictions to be a smooth function is a way to control for overfit. We are here particularly interested in regression problems with regressors constrained to limited regions in the regressor-space *e.g.,* a manifold. For this type of systems we develop a new regression technique, *Weight Determination by Manifold Regularization* (WDMR). WDMR is inspired by applications in biology and developments in manifold learning and uses regularization for smoothness to obtain smooth estimates. The use of regularization for smoothness in linear system identification is also discussed.

The thesis also presents a real-time *functional Magnetic Resonance Imaging* (fMRI) bio-feedback setup. The setup has served as proof of concept and been the foundation for several real-time fMRI studies.

Populärvetenskaplig sammanfattning

Modeller används inom de flesta områden för att efterlikna verkligheten. Anledningarna kan vara allt ifrån att det är fysikaliskt omöjligt till att det är kostsamt att utföra experimenten och därför utförs dessa på en modell istället. En modell kan också användas till att generalisera och förutse beteenden för nya situationer. Vi använder exempelvis en mental modell för cykling för att från tidigare erfarenheter kunna hantera nya situationer.

I denna avhandling studeras matematiska modeller. Framför allt diskuteras en teknik för att framkalla egenskaper så som *gleshet* och *glatthet* hos modellparametrar och skattningar. Denna teknik betecknas *regularisering*. Varför är man då intresserad av att framkalla dessa egenskaper? Gleshet kan vara av nytta för att välja ut mätstorheter som man bör fortsätta att mäta om man vill bibehålla goda skattningsresultat. Om det är kostsamt att mäta kan denna användning vara värdefull. Gleshet har också visats användbart vid medicinsk bildbehandling för till exempel minskning av röntgentider. I denna avhandling används regularisering för gleshet på problem inom områdena *systemidentifiering* och *signalbehandling*. Bland annat diskuteras hur regularisering för gleshet kan användas för att upptäcka plötsliga förändringar. Glatthet är i många fall motiverat av fysikaliska skäl. Många signaler som är intressanta att modellera och förutse beter sig på ett mjukt och kontinuerligt sätt. Det finns därför skäl till att modellen som används även har dessa egenskaper. Ett av resultaten i denna avhandling är en ny modelleringsmetod, *Weight Determination by Manifold Regularization* (WDMR). Ett specifikt användningsområde som diskuteras är skattning av vattentemperatur från mätningar av den kemiska sammansättningen i musselskal. Antagandet att det finns ett glatt samband mellan den kemiska koncentrationen i musselskalet och temperaturen är här viktigt för bra skattningar.

Ett annat område som berörs i avhandlingen är mätning av hjärnaktivitet. Mer specifikt presenteras en praktisk uppställning för att mäta och tolka hjärnaktivitet i realtid.

Acknowledgments

I have enjoyed my PhD studies a lot! There are a number of reasons for that. First of all, I have had a great supervisor. My supervisor, Professor Lennart Ljung, has guided and inspired me through out the years of my PhD. Thank you Lennart, you been outstanding! Dr. Jacob Roll, my assistant supervisor, has also been of great importance. I am very grateful for all our discussions and for all the help you given me. Ulla Salaneck and Åsa Karmelind have also been invaluable. Thank you!

Secondly, the automatic control group at Linköping University is beyond ordinary. It is a creative, friendly, environment, and an excellent place for PhD studies. In particular I would like to thank Dr. Umut Orguner for all your help, interesting discussions and for being so kind! Also, thank you Dr. Tianshi Chen for interesting discussions and a nice collaboration. Thank you Lic Christian Lundquist, Dr. Ragnar Wallin, Zoran Sjanic, Lic Christian Lyzell, Daniel Petersson, Karl Granström, Lic Daniel Ankelhed and Patrik Axelsson for all the time away, moules frites, early mornings, balconies and Kalles kaviar. Windsurfing and kite people, Dr. Henrik Tidefelt, Dr. Johan Sjöberg, Dr. David Törnqvist, Tohid Ardeshiri, André Carvalho Bittencourt, Fredrik Lindsten, Dr. Emre Özkan and Sina Khoshfetrat Pakazad, it has been lots of fun! Office mates, Lic Johanna Wallén and Jonas Callmer, thanks a lot for your company and happy Fridays! Thank you Dr. Thomas Schön and Dr. Gustaf Hendeby for your company at Campushallen. Professor Fredrik Gustafsson and Professor Torkel Glad, thank you for great courses and collaborations!

External collaborators, thank you to Professor Anders Ynnerman, Professor Hans Knutsson, Dr. Mats Andersson, Dr. Joakim Rydell, Dr. Anders Brun, Lic Anders Eklund and Tan Khoa Nguyen for the collaboration within the MOVIII project. Thank you Dr. Carl Edward Rasmussen at University of Cambridge for a very nice stay in Cambridge. Professor Stephen Boyd, Maite Bauwens, Dr. Marc Deisenroth and Tillmann Falck, thank you for interesting discussions and good collaborations.

This thesis has been proofread by Professor Lennart Ljung, Patrik Axelsson, Daniel Petersson, Dr. David Törnqvist, Dr. Umut Orguner, Dr. Mehmet Guldogan, sister Pernilla Ohlsson and Dr. Thomas Schön. Thank you for your comments! Also thanks to Dr. Gustaf Hendeby, Dr. Henrik Tidefelt and Dr. David Törnqvist for LaTeX support.

Noelia! You been and are my love. You made me laugh and you made me happy. Thanks a lot for your patience! My family gets a lot of love and gratefulness too. You have always been there for me, even though I have been away. Thank you! Also thank you to friends from Uppsala, Amherst, Linköping, Y2000d, Cambridge for many happy memories!

I am also very grateful for the support from the Strategic Research Center MOVIII and from the Swedish Research Council in the Linnaeus center CADICS. It has been very motivating and I am very glad to have gotten the opportunity to be a part of these research centers.

Linköping, October 2010
Henrik Ohlsson

Contents

II Publications

Notation

Mathematical Symbols

Notation	Meaning
$\mathcal{R}$	set of real numbers
$\mathcal{R}^+$	set of positive real numbers
$\mathcal{Z}$	set of integers
$\mathcal{N}$	set of natural numbers
$\|\cdot\|_p$	ℓ_p-norm
$\|\cdot\|$	absolute value for a scalar and the determinant for a matrix
$dim(x)$	dimension of the vector x
$card(\mathcal{X})$	cardinality of the set $\mathcal{X}$
$rank(X)$	column rank of the matrix X
$sign(\cdot)$	sign function
$tr(\cdot)$	trace
I_n	$n \times n$-dimensional identity matrix
$0_{n\times m}$	$n \times m$-dimensional zero matrix
$1_{n\times m}$	$n \times m$-dimensional matrix of ones
$N(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2
$N(x; \mu, \sigma^2)$	Gaussian distribution in x with mean μ and variance σ^2
$U(a, b)$	uniform distribution between a and b
$p_e(\cdot)$	probability distribution for e
$\{x_t\}_{t=1}^N$	set containing $x_1, x_2, \ldots, x_N$
E_x	expectation with respect to the random variable x
$\triangleq$	equal by definition
$\in$	belongs to
X^T	transpose of the matrix X
$\dot{x}$	time derivative of x
∇_θ	gradient with respect to θ
$\emptyset$	empty set

$\bigcap$	intersection
$\subset$	proper subset
$\subseteq$	subset
∂_x	subdifferential with respect to x
$k(\cdot, \cdot)$	kernel
ℓ	length-scale of a squared exponential kernel
φ	regressor
y	system output
u	system input
x	state
λ	regularization parameter
$\mathcal{N}_{\mathsf{o}}$	index set associated with the observed data
$\mathcal{N}_{\mathsf{e}}$	index set associated with the estimation-data set
N_{e}	number of elements in the estimation-data set
$\mathcal{N}_{\mathsf{v}}$	index set associated with the validation-data set
N_{v}	number of elements in the validation-data set
$\mathcal{N}_{\mathsf{t}}$	index set associated with the test-data set
N_{t}	number of elements in the test-data set
$f(\varphi, \theta)$	model evaluated at the regressor φ and the regressor parameter θ
$f(\varphi)$	model evaluated at the regressor φ
f_0	system function
T_s	sample time

Abbreviations and Acronyms

Abbreviation	Meaning
AFMM	Adaptive Forgetting by Multiple Models
AIC	Akaike Information Criterion
ARX	Auto-Regressive with eXogenous variables
BCI	Brain Computer Interface
BLUE	Best Linear Unbiased Estimator
BOLD	Blood Oxygen Level Dependent
CCA	Canonical Correlation Analysis
CS	Compressed Sensing
CUSUM	CUmulative SUM
CV	Cross Validation
EKF	Extended Kalman Filter
FDI	Fault Detection and Isolation
FIR	Finite Impulse Response
fMRI	functional Magnetic Resonance Imaging
FOCUSS	FOCal Underdetermined System Solver
GLM	General Linear Modeling
GP	Gaussian Process
GPCA	General Principal Component Analysis

GPR	Gaussian Process Regression
HMM	Hidden Markov Model
i.i.d.	independent and identically distributed
IMM	Interacting Multiple Model
KF	Kalman Filter
KKT	Karush-Kuhn-Tucker
K-NN	K-Nearest Neighbor
LARS	Least Angle Regression
lasso	least absolute shrinkage and selection operator
LLE	Locally Linear Embedding
LQG	Linear-Quadratic-Gaussian
LS	Least-Squares
LS-SVM	Least-Squares Support Vector Machines
LS-SVR	Least-Squares Support Vector Regression
MAE	Mean Absolute Error
MAP	Maximum A Posteriori
MLE	Maximum Likelihood Estimate
MPC	Model Predictive Control
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
OE	Output Error
PCA	Principal Component Analysis
PEM	Prediction Error Method
PLS	Partial Least Squares
PRBS	Pseudo-Random Binary Sequence
PWA	Piece-Wise Affine
PWARX	Piece-Wise Auto-Regressive with eXogenous variables
PWASON	Piece-Wise Affine system identification using Sum-Of-Norms regularization
RKHS	Reproducing Kernel Hilbert Space
SISO	Single-Input Single-Output
SNR	Signal-to-Noise Ratio
s.t.	subject to
STATESON	STATE estimation by Sum-Of-Norms regularization
SVM	Support Vector Machines
SVR	Support Vector Regression
UAV	Unmanned Aerial Vehicle
UTM	Universal Transverse Mercator
WDMR	Weight Determination by Manifold Regularization
w.p.	with probability
w.r.t.	with respect to

Part I

Background

1 Introduction

1.1 Models and Modeling

Models are used in most scientific disciplines as substitutes for reality. It can be that it is practically impossible to conduct experiments on the physical system and a model thereof is therefore used to replace it. Or it could be that the model is used to generalize to new situations not previously seen.

We humans use models every day, *mental models*. These models are built-up from past experiences and make it possible for us to, *e.g.,* ride our bikes. When we bike, we use our mental model for biking to not fall over. In particular, we need to use previous biking experience to generalize to new situations.

In this thesis, methods for computing models are discussed. Like for a human, most of the models will be based on gathered past observations. We do not summarize these in a mental model, but seek instead a *mathematical model* that can explain these observations. A mathematical model describes a system's behavior using mathematical language. Mathematical language could be a set of differential or difference equations, or it could be a rule for how to combine past observations.

Mental models are of particular use for us and our brain. Mathematical models are not useful for our brain (at least not in the same way as mental models) but of particular interest and use for engineering and science. The two next examples motivate the use of mathematical models. We will return to both of these examples at later phases of this thesis.

Example 1.1: Climate Reconstruction

There exist a number of climate recorders in nature from which the past temperature can be extracted. However, only a few natural archives are able to record climate fluctuations with high enough resolution so that the seasonal variations can be reconstructed. One such archive is a bivalve shell, see Figure 1.1. The chemical composition of a shell of a bivalve depends on a number of chemical and physical parameters of the water in which the shell was composed. Of these parameters, the water temperature is probably the most important one. It should therefore be possible to estimate the water temperature for the time the shell was built, from measurements of the shell's chemical composition. This would *e.g.,* give climatologists the ability to estimate past water temperatures by analyzing ancient shells. To do this, a model for how the chemical composition relates to water temperature would be needed.

Figure 1.1: *Bivalve shell.*

Example 1.2: Model-Based Reference Generation

Flight planning is essential for safety when flying. It makes sure that, on the flight route, the airplane does not get to close to other airplanes, takes into account weather forecasts, fuel consumption and time constraints, and makes sure that the airplane reaches its final destination. A route, in its simplest form, is a set of ordered coordinates, *waypoints*. In an autopilot of a commercial airplane or in the computer of an *Unmanned Aerial Vehicle* (UAV), waypoints are used to generate reference trajectories which the controllers then use to navigate between the waypoints. The most primitive *reference generator* does not take into account limitations and the dynamics of the airplane. It gives a reference which is simply a sequence of line segments connecting the waypoints. The airplane will not be able to follow this reference very well and it is obvious that fuel could have been saved and the comfort of the passengers could have been improved if instead a smooth trajectory would have been generated. However, any smooth trajectory does not suffice. The airplane may *e.g.,* be too large to follow the turns which may cause a not so smooth behavior after all. Therefore, a better approach would be to include a model of the airplane in the reference generator and do a *model-based reference generation*.

Model-based reference generation is a particular type of *trajectory generation* and of interest for *e.g.,* industrial robotics and planning for unmanned vehicles. Trajectory generation is further discussed in Paper D in Part II.

Since mathematical models are used and of importance in so many different fields, there are of course a huge variety of different types of models and modeling techniques. There are also several fields studying the act of modeling, each with its own nomenclature. In *system identification e.g.,* the act of modeling is referred to as *identification* and in the closely related field of *machine learning*, the

term *learning* or *inference* is used. Since this is a thesis in system identification, we will most of the time stick to the nomenclature used there.

Mathematical modeling can be divided into two categories. Modeling either belongs to *regression* or *classification*. In this thesis we are only concerned with regression. There is further a focus on different types of regularizations. This is also reflected in the name of the thesis.

1.2 Regularization

Regularization is a methodology for making an *ill-posed* problem *well-posed* (Poggio et al., 1985; Neumaier, 1998). A problem is ill-posed (Hadamard (1902), see also Tikhonov and Arsenin (1977, p. 7)) if its solution

- does not exist,
- is not unique or
- does not depend continuously on the input data.

If a problem is not ill-posed, it is well-posed. An example of an ill-posed problem could be the task of finding *the* $x \in \mathcal{R}^{n_x}$, given $y \in \mathcal{R}^{n_y}$ and $A \in \mathcal{R}^{n_y \times n_x}$, that solves

$$\min_x \|y - Ax\|_2^2. \tag{1.1}$$

If $rank(A) < n_x$ the minimizing x is not unique and the problem hence ill-posed. A well-posed regularized version of the problem is given by the regularized least squares problem

$$\min_x \|y - Ax\|_2^2 + \lambda \|x\|_2^2, \quad \lambda \in \mathcal{R}^+. \tag{1.2}$$

The added term $\lambda \|x\|_2^2$ conveys the desire that $\|x\|_2^2$ should be small. It also makes the solution unique and the problem well-posed. Regularization can also be used to communicate other prior thoughts concerning a parameter, signal or model. Common properties imposed by regularization are smoothness or sparseness, as we will see later. We will return to the regularized least squares problem in later chapters and leave the details for then.

Regularization is also a way to control for *overfitting*. Overfitting is a problem that can occur in the estimation process of a model and in particular when a stochastic noise process is modeled as a deterministic signal. The most common way to avoid overfitting is to limit the model's ability to pick up rapid variations in the data, often associated with the noise. One technique for doing this is regularization. By controlling for overfitting a *bias* is usually introduced. The *variance* is however decreased. Regularization is therefore also a way to deal with the *bias-variance trade-off* for a model.

In statistics and machine learning, regularization has gained popularity due to modeling methods such as *Support Vector Machines* (SVM, Vapnik (1979, 1995)), *ridge regression* (Hoerl and Kennard (1970), see also Hastie et al. (2001), p. 59)

and *lasso* (least absolute shrinkage and selection operator, Tibsharani (1996), see also Hastie et al. (2001), p. 64). When the *Bayesian* approach to modeling is used, regularization often shows up and can be associated with the prior knowledge.

In system identification, the *Akaike Information Criterion* (AIC, Akaike (1973)) is a well known way to balance the model fit against the model complexity. Regularization here acts as a price on model complexity.

Regularization has also had a great impact on many applications, and very much so in clinical imaging. In *e.g.,* breast cancer imaging, the number of sensors is physically restricted which leads to long scan times. Regularization and sparsity can be used to reduce that, as shown in Guo et al. (2010) and Brady et al. (2009). In *Magnetic Resonance Imaging* (MRI), the number of scans is physically limited and to obtain high resolution images, regularization plays a key role, see *e.g.,* Brady et al. (2009).

Example 1.3: Compressed Sensing

The *Nyquist-Shannon sampling criterion* states that for a *bandlimited* (no energy above a certain frequency) signal, the sampling frequency should be twice that of the bandlimit to guarantee the possibility to perfectly reconstruct the time-continuous signal (see *e.g.,* Oppenheim et al. (1996, p. 519)). That means that to obtain a (good) audio recording a sampling frequency of at least 40 kHz is needed, since our ears are sensitive to frequencies up to 20 kHz. However, MP3 files are often around 3 megabytes, not 30 megabytes (a three minute stereo recording gives $3 \cdot 60 \cdot 2 \cdot 40 \cdot 10^3 = 14.4 \cdot 10^6$ samples. A precision of 16 bits gives 28.8 megabytes). Data compression is of course the reason for this storage saving. A sound is hence sampled, stored and then compressed. In the compression, about 90% of the storage area is returned.

It may seem meaningless to measure a lot of information if 90% will be thrown away before someone even listened to the song. Since this thesis is about regularization, you may guess that regularization can help to sample more efficiently. And yes, a regularization technique called *Compressed Sensing* (CS, Donoho (2006); Candès et al. (2006)) is what is needed.

We continue and reveal the details behind compressed sensing in Chapter 4. An interesting and well written paper on compressed sensing which inspired to above example is given by Hayes (2009).

1.3 State Estimation

Dynamic systems are characterized by that their output depends on current and past inputs. The effect that these inputs have had on the system is gathered in the *state*. The state contains valuable information for *e.g.,* controllers and for decision making. The state is however often not directly measurable. It is therefore of interest to be able to estimate the state using the available measurements. The theory for doing this is called *state estimation*.

The focus of this thesis is not state estimation. A brief description necessary to understand the paper on state estimation in Part II is therefore only provided. In particular we discuss state estimation under process noise which is often zero but occasionally non-zero, leading to so called *load disturbances*.

1.4 Notation

It is strategic, before readers detach and jump to chapters of their choice, to explain some notational choices made throughout the thesis. Lower-case letters will be used for scalars and column vectors, while upper-case letters are used to denote matrices. "$(\cdot)$" will be used to pick out elements of vectors or matrices. $x(t)$ hence denotes the tth element of the vector x. ":" will be used, as in MATLAB, to pick-out a sequence of elements of a matrix or vector. $A(1:2,:)$ hence denotes the two top rows of the matrix A. Calligraphic letters will be used for sets. Models will be denoted by $f(\varphi,\theta)$, φ being a regressor and θ the model-parameters. $f_0(\varphi)$ will be used to denote the true system that we try to imitate using a model. "$\hat{}$" denotes an estimate of some quantity. $\hat{x}$ therefore denotes an estimate of x. A subscript will be used to index time or as sample index. x_t hence denotes the variable x at time or index t. In some papers of Part II, "$(\cdot)$" is used instead of subscript. Some exceptions to these notational choices exist.

See also listed mathematical symbols and abbreviations on pages xvii and xviii.

1.5 Publications

Published work of relevance to this thesis is listed below in chronological order. Publications marked with a "$*$" are included in Part II of this thesis.

H. Ohlsson, J. Roll, T. Glad, and L. Ljung. Using manifold learning for nonlinear system identification. In *Proceedings of the 7th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, Pretoria, South Africa, August 2007.

H. Ohlsson. *Regression on manifolds with implications for system identification*. Licentiate thesis no. 1382, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, December 2008.

H. Ohlsson, J. Roll, A. Brun, H. Knutsson, M. Andersson, and L. Ljung. Direct weight optimization applied to discontinuous functions. In *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008a.

H. Ohlsson, J. Roll, and L. Ljung. Manifold-constrained regressors in system identification. In *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008b.

* H. Ohlsson, J. Rydell, A. Brun, J. Roll, M. Andersson, A. Ynnerman, and H. Knutsson. Enabling bio-feedback using real-time fMRI. In *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008c.

A. Eklund, H. Ohlsson, M. Andersson, J. Rydell, A. Ynnerman, and H. Knutsson. Using real-time fMRI to control a dynamical system. In *Proceedings of the 17th Meeting of the International Society for Magnetic Resonance in Medicine (ISMRM)*, Honolulu, USA, April 2009a.

M. Bauwens, H. Ohlsson, K. Barbé, V. Beelaerts, J. Schoukens, and F. Dehairs. Three ways to do temperature reconstruction based on bivalve-proxy information. In *Proceedings of the 28th Benelux Meeting on Systems and Control*, Spa, Belgium, March 2009b.

H. Ohlsson and L. Ljung. Gray-box identification for high-dimensional manifold constrained regression. In *Proceedings of the 15th IFAC Symposium on System Identification, SYSID 2009*, Saint-Malo France, July 2009.

M. Bauwens, H. Ohlsson, K. Barbé, V. Beelaerts, F. Dehairs, and J. Schoukens. On climate reconstruction using bivalve shells: Three methods to interpret the chemical signature of a shell. In *Proceedings of the 7th IFAC Symposium on Modelling and Control in Biomedical Systems*, Aalborg, Denmark, August 2009a.

A. Eklund, H. Ohlsson, M. Andersson, J. Rydell, A. Ynnerman, and H. Knutsson. Using real-time fMRI to control a dynamical system by brain activity classification. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'09)*, London, UK, September 2009b.

H. Ohlsson, M. Bauwens, and L. Ljung. On manifolds, climate reconstruction and bivalve shells. In *Proceedings of the 48th IEEE Conference on Decision and Control*, Shanghai, China, December 2009.

F. Lindsten, J. Callmer, H. Ohlsson, D. Törnqvist, T. B. Schön, and F. Gustafsson. Geo-referencing for UAV navigation using environmental classification. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, May 2010.

K. Nguyen, A. Eklund, H. Ohlsson, F. Hernell, P. Ljung, C. Forsell, M. Andersson, H. Knutsson, and A. Ynnerman. Concurrent volume visualization of real-time fMRI. In *Proceedings of the IEEE International Symposium on Volume Graphics 2010*, Norrköping, Sweden, May 2010.

* H. Ohlsson, L. Ljung, and S. Boyd. Segmentation of ARX-models using sum-of-norms regularization. *Automatica*, 46(6):1107–1111, 2010d.

A. Eklund, M. Andersson, H. Ohlsson, A. Ynnerman, and H. Knutsson. A brain computer interface for communication using real-time fMRI. In *Proceedings of the International Conference on Pattern Recognition 2010*, Istanbul, Turkey, August 2010.

H. Ohlsson and L. Ljung. Semi-supervised regression and system identification. In X. Hu, U. Jonsson, B. Wahlberg, and B. Ghosh, editors, *Three Decades of Progress in Control Sciences*. Springer Verlag, December 2010a. To appear.

H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. State smoothing by sum-of-norms regularization. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, USA, December 2010a. To appear.

* H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. Trajectory generation using sum-of-norms regularization. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, USA, December 2010b. To appear.

T. Chen, T. B. Schön, H. Ohlsson, and L. Ljung. Decentralization of particle filters using arbitrary state partitioning. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, USA, December 2010a. To appear.

M. Bauwens, H. Ohlsson, K. Barbé, V. Beelaerts, F. Dehairs, and J. Schoukens. On climate reconstruction using bivalves: Three methods to interpret the chemical signature of a shell. *Computer Methods and Programs in Biomedicine*, 2010a. Accepted for publication.

M. Bauwens, H. Ohlsson, K. Barbé, V. Beelaerts, F. Dehairs, and J. Schoukens. A nonlinear multi-proxy model based on manifold learning to reconstruct water temperature from high resolution trace element profiles in biogenic carbonates. *Geoscientific Model Development*, 2010b. Accepted for publication.

T. Chen, T. B. Schön, H. Ohlsson, and L. Ljung. Decentralized particle filter with arbitrary state partitioning. *IEEE Transactions on Signal Processing*, 2010b. Accepted for publication.

* H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. State estimation under abrupt changes using sum-of-norms regularization. *Automatica*, 2010c. Submitted, under revision.

* H. Ohlsson and L. Ljung. Weight determination by manifold regularization. In *Distributed Decision-Making and Control*, Lecture Notes in Control and Information Sciences. Springer Verlag, 2010b. Submitted.

* H. Ohlsson and L. Ljung. Piecewise affine system identification using sum-of-norms regularization. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011. Submitted.

* T. Chen, H. Ohlsson, and L. Ljung. On the estimation of transfer functions, regularizations and Gaussian processes – Revisited. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011. Submitted.

  T. Falck, H. Ohlsson, L. Ljung, J. A.K. Suykens, and B. De Moor. Segmentation of times series from nonlinear dynamical systems. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011. Submitted.

  M. P. Deisenroth and H. Ohlsson. General perspective to Gaussian filtering and smoothing: Explaining current and deriving new algorithms. In *Proceedings of the American Control Conference (ACC), 2011*, San Francisco, USA, 2011. Submitted.

1.6 Contributions

Sparseness has had a huge impact on neighboring scientific disciplines, such as machine learning and signal processing, but has had very little effect on system identification. One of the major contributions of this thesis is therefore the new developments in system identification using sparsity. Relevant readings are Papers A and B in Part II of this thesis. See also related contributions in signal processing, Papers C and D.

Manifold learning, unsupervised learning and semi-supervised learning are well establish areas in machine learning. In system identification, these subjects have hardly been given any consideration at all. A contribution of this thesis is therefore the increased understanding for these subjects and how they can be of use in system identification. Relevant reading is Paper E in Part II of this thesis.

The author of this thesis has also carried out research in *functional Magnetic Resonance Imaging* (fMRI). This contribution is described in Paper G in Part II of this thesis.

1.7 Thesis Outline

The thesis is divided into two parts. The first part contains motivations and background theory and the second part a collection of papers.

1.7.1 Outline of Part I

Chapter 2 serves as an introduction to mathematical modeling and regression and introduces the fundamental knowledge and the necessary notation for the subsequent chapters. Readers familiar with the subject can skip this chapter. Chapter 3 gives a brief introduction to state estimation. Chapter 4 discusses regularization for sparseness and Chapter 5 discusses regularization for smoothness. The last chapter of Part I gives a conclusion and discusses interesting future research directions.

1.7.2 Outline of Part II

Part II presents a collection of papers that is relevant for the thesis.

The four first papers further develop the theory presented in Chapters 3 and 4. **Paper A**,

> H. Ohlsson, L. Ljung, and S. Boyd. Segmentation of ARX-models using sum-of-norms regularization. *Automatica*, 46(6):1107–1111, 2010d.

discusses what sparseness and segmented ARX models have in common. A new approach using regularization to estimate segmented ARX models is presented. The author of this thesis was the major contributor in writing this paper and in the research prior the paper. The author of this thesis also came up with the idea of using regularization for sparseness in the estimation of segmented ARX models. This paper inspired to several other applications of regularization for sparseness, see *e.g.,* Ohlsson et al. (2010a,b,c); Ohlsson and Ljung (2011); Falck et al. (2011). This work also initialized collaboration with Professor Stephen Boyd at Stanford University.

Paper B,

> H. Ohlsson and L. Ljung. Piecewise affine system identification using sum-of-norms regularization. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011. Submitted.

extends the theory presented in Paper A to piecewise affine systems. A regularization approach is again taken. The author of this thesis was the major contributor in writing the paper and in the research prior the paper. The author of this thesis also came up with the idea of using regularization for sparseness in piecewise affine system identification.

Paper C,

> H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. State estimation under abrupt changes using sum-of-norms regularization. *Automatica*, 2010c. Submitted, under revision.

discusses how sparseness can help in state estimation when abrupt changes are present, *e.g.,* load disturbances. The author of this thesis was the major contributor in writing the paper and in the research prior the paper. It was Professor

Lennart Ljung's idea to use regularization for sparseness together with state estimation. Parts of the theory presented in this paper have also been presented in Ohlsson et al. (2010a).

Paper D,

> H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. Trajectory generation using sum-of-norms regularization. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, USA, December 2010b. To appear.

presents a model-based trajectory generation scheme. Sparsity and regularization are here used to give a compact representation for the trajectory, something that is desired when communication and storage are limited. The author of this thesis was the major contributor in writing the paper and in the research prior the paper. It was Professor Fredrik Gustafsson's idea to use regularization for sparseness for trajectory generation.

The fifth paper, **Paper E**,

> H. Ohlsson and L. Ljung. Weight determination by manifold regularization. In *Distributed Decision-Making and Control*, Lecture Notes in Control and Information Sciences. Springer Verlag, 2010b. Submitted.

discusses a novel regression method *Weight Determination by Manifold Regularization* (WDMR). The regression method has strong bounds with manifold learning and has inherited properties thereof. Unlike most methods in system identification, WDMR is a semi-supervised regression method. WDMR uses regularization to control for smoothness and is therefore related to theory developed in Chapter 5. The author of this thesis was the major contributor in writing the paper and in the research prior the paper. A pre-study was presented in Ohlsson et al. (2007). WDMR, in its present formulation, was first presented in Ohlsson et al. (2008b). A number of interesting applications and extensions of WDMR have also been presented, *e.g.,* Ohlsson (2008); Ohlsson and Ljung (2009). The application to temperature reconstruction from bivalves is probably the most exciting, see *e.g.,* Ohlsson et al. (2009); Bauwens et al. (2009a, 2010a,b). The work behind WDMR has led an extensive collaboration with researchers at Vrije Universiteit Brussel. The author of this thesis came up with the idea behind WDMR.

Paper F,

> T. Chen, H. Ohlsson, and L. Ljung. On the estimation of transfer functions, regularizations and Gaussian processes – Revisited. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011. Submitted.

continues the discussion of regularization for smoothness and examines how regularization can be used in linear system identification. The theory presented in Paper F is also related to theory developed in Chapter 5. The author of this thesis

was an active contributor in the work prior writing the paper and in writing the paper.

Paper G,

> H. Ohlsson, J. Rydell, A. Brun, J. Roll, M. Andersson, A. Ynnerman, and H. Knutsson. Enabling bio-feedback using real-time fMRI. In *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008c.

presents a real-time fMRI bio-feedback setup. fMRI is a method for measuring brain activity. The conventional use of fMRI is in "batch-mode". The subject is first scanned for 30 minutes. Then the data is analyzed and brain activity detected and located using smoothing on the batch of fMRI measurements. The setup presented here hence presents a real-time fMRI setup *i.e.,* fMRI measurements are analyzed as they are acquired. The setup presented led the way for several interesting real-time fMRI studies *e.g.,* Eklund et al. (2009a,b, 2010); Nguyen et al. (2010) and shows some more applied research conducted by the author of this thesis. The author of this thesis was the main contributor to the presented setup.

2

Mathematical Modeling and Regression

Models summarize available knowledge about the system. Available knowledge can be physical first principles describing the behavior of the system or it can be measurements of system specific quantities.

2.1 Types of Models and Modeling

When only physical first principles are used, modeling, or the act of finding a model, is referred to as *white-box modeling*. When modeling is solely based on measurements it is referred to as *black-box modeling* and when physical principles are combined with measurements, *gray-box modeling*.

A model (and also a system) is either *dynamic* or *static*. The output of a dynamic model depends on previous and current system inputs, while a static model only depends on the system input at the moment. One may say that a static model is memoryless, while a dynamic model contains a memory in which past inputs are stored. The words "dynamical" and "dynamic" are used interchangeably in the literature.

A *model* is made up of a *model structure* and a set of *model parameters*. Model parameters are quantities that are chosen to make the model imitate the specific system under consideration. For example, a mass-spring system can readily be modeled by a second order differential equation

$$\frac{d^2x_t}{dt^2} + a\frac{dx_t}{dt} + bx_t = c \tag{2.1}$$

in the position x of the mass. To make the model imitate a specific mass-spring system, the model parameters a, b and c have to be set. This could *e.g.,* be done

by comparing predicted mass positions of the model with observed positions. A second order differential equation is the model structure in this case and the coefficients a, b and c, the model parameters. For the second order differential equation model, the number of parameters is fixed and equal to three. That the number of model parameters is fixed characterizes a *parametric model*. The number of parameters of a *non-parametric model* typically grows with the number of observations available for estimating the model. It may seem a bit counter intuitive that a non-parametric model has parameters and often considerably more parameters than a parametric model, but that is the convention.

The quantity of interest can either belong to a set of a finite number of elements, and is then said to be *qualitative*. When the quantities are qualitative they are often denoted *labels* and the act of modeling, *classification*. Or, if on the other hand, the quantity of interest can take any value in *e.g.,* an interval, the act of modeling is referred to as *regression*. The considered quantities are then said to be *quantitative*. This thesis only treats quantitative quantities and the regression problem.

It is also common to separate a *Bayesian* approach to modeling from a non-Bayesian approach, sometimes called a *frequentist's* or a *classical* approach. Sections 2.4 and 2.5 take a non-Bayesian approach and Section 2.9 discusses a Bayesian approach to modeling.

2.2 The Regression Problem

Many problems in estimation and identification can be formulated as regression problems. In a regression problem we are seeking to determine the relationship between a *regression vector* φ (input, independent variable) and a quantity of interest, a quantitative variable y (output, dependent variable), here called the *output*. Basically this means that we would like to find the function f_0 that describes the relationship

$$y = f_0(\varphi). \tag{2.2}$$

With $\varphi \in \mathcal{R}^{n_\varphi}$ and $y \in \mathcal{R}$, f_0 is a mapping from $\mathcal{R}^{n_\varphi} \to \mathcal{R}$. For simplicity, $y \in \mathcal{R}$ will be assumed throughout the rest of this chapter.

Measuring always introduces some uncertainty, which motives the introduction of a discrepancy or noise term e,

$$y = f_0(\varphi) + e. \tag{2.3}$$

This implies that there is no longer a unique y corresponding to a φ. We will assume that the noise sequence $\{e\}$ obtained as f_0 is measured multiple times is constructed from *independent and identically distributed* (i.i.d.) zero mean stochastic variables. Let further p_e be the probability distribution associated with the random variable e.

In practice our estimate of $f_0(\varphi)$ has to be computed from a limited number of

observations of (2.3). The problem is hence to observe a number of connected pairs $\{\varphi, y\}$, and then based on this information be able to provide a guess or estimate for f_0 that is related to any given, new, value of φ.

The estimate of f_0, or the model, that we choose to work with can either be *linear* or *nonlinear*. For a linear model, the model output is a linear function of the regressors while for a nonlinear model, the model output is allowed to be a nonlinear function of the regressors.

2.3 Estimation, Validation and Test Data

Given a set of observations, $\{(\varphi_t, y_t)\}_{t\in\mathcal{N}_\mathsf{o}}$, $\mathcal{N}_\mathsf{o} \subset \mathcal{Z}$, it is often a good idea to separate the observation data set into three sets:

- The *estimation data* set is used to compute the model, *e.g.*, to compute the model parameters in a parametric model. The estimation data set will be denoted by $\{(\varphi_t, y_t)\}_{t\in\mathcal{N}_\mathsf{e}}$, $\mathcal{N}_\mathsf{e} \subseteq \mathcal{N}_\mathsf{o}$. Let also $N_\mathsf{e} \triangleq card(\mathcal{N}_\mathsf{e})$.
- The *validation data* set is used to examine an estimated model's ability to predict the output of a new set of regressor data. Having a number of prospective models of different structures, the validation data can be utilized to choose the best performing model structure. For example the number of delayed system inputs and outputs used in the regressors in a parametric model could be chosen using the validation data. The validation data set will be denoted by $\{(\varphi_t, y_t)\}_{t\in\mathcal{N}_\mathsf{v}}$, $\mathcal{N}_\mathsf{v} \subseteq \mathcal{N}_\mathsf{o}$, $\mathcal{N}_\mathsf{v} \bigcap \mathcal{N}_\mathsf{e} = \emptyset$. Let also $N_\mathsf{v} \triangleq card(\mathcal{N}_\mathsf{v})$. How the validation data is used is discussed in Section 2.5.
- The *test data* set is used to test the ability of the chosen model (with the parameter choice from the estimation step and the structure choice from the validation step) to predict new outputs. The test data set can be used to gain confidence for the chosen model. The test data set will be denoted by $\{(\varphi_t, y_t)\}_{t\in\mathcal{N}_\mathsf{t}}$, $\mathcal{N}_\mathsf{t} \subseteq \mathcal{N}_\mathsf{o}$, $\mathcal{N}_\mathsf{t} \bigcap \mathcal{N}_\mathsf{e} = \emptyset$, $\mathcal{N}_\mathsf{t} \bigcap \mathcal{N}_\mathsf{v} = \emptyset$. Let also $N_\mathsf{t} \triangleq card(\mathcal{N}_\mathsf{t})$.

2.4 Fitting a Model

Having divided the observations into an estimation, validation and test data set, we are ready to estimate a model. The conventional approach within system identification is to make use of a parametric model $f(\varphi_t, \theta)$, which is hopefully flexible enough to imitate the transformation f_0 in (2.3). Here θ is used to denote the model parameters. Examples of structures that will be used in this thesis are:

- The *Auto-Regressive with eXogenous variables* (ARX) model structure. This structure leads to a linear model. If we consider a single-input single-output dynamic system with the input u_t and the output y_t, the ARX model takes the form

$$f(\varphi_t, \theta) = \varphi_t^T \theta, \quad \varphi_t = \begin{bmatrix} -y_{t-1} & \dots & -y_{t-na} & u_{t-1} & \dots & u_{t-nb} \end{bmatrix}^T. \tag{2.4}$$

The quantities na and nb are parameters of the model structure.

- The *Finite Impulse Response* (FIR) model structure. This structure also leads to a linear model. If we again let u_t be an input of a single-input single-output dynamic system, the FIR model takes the form

$$f(\varphi_t, \theta) = \varphi_t^T \theta, \quad \varphi_t = \begin{bmatrix} u_{t-1} & \dots & u_{t-nb} \end{bmatrix}^T. \tag{2.5}$$

nb is the *order* of the FIR model.

For more on the model structures briefly introduced above, and several other model structures used in system identification, see *e.g.,* Ljung (1999, Chap. 4).

$f(\varphi_t, \theta)$ is adjusted to the regressor-output pairs of the estimation data set $\{(\varphi_t, y_t)\}_{t \in \mathcal{N}_e}$ by choosing θ as

$$\hat{\theta} = \arg\min_{\theta} \sum_{t \in \mathcal{N}_e} l\Big(y_t - f(\varphi_t, \theta)\Big), \tag{2.6}$$

where $l : \mathcal{R} \to \mathcal{R}$ is a function of the *prediction error* $y_t - f(\varphi_t, \theta)$ and typically chosen as a norm. In system identification, the use of (2.6) to estimate a model parameter is a special case of the *Prediction Error Method* (PEM, see *e.g.,* Ljung (1999, 2002)). Also, if we set l as the negative logarithm of the measurement noise distribution, *i.e.,* $l(\cdot) = -\log p_e(\cdot)$, then $\hat{\theta}$ of (2.6) equals the *Maximum Likelihood Estimate* (MLE) of θ (see *e.g.,* Ljung (2002)).

With measurement noise present, obtaining a perfect fit *i.e.,*

$$\sum_{t \in \mathcal{N}_e} l\Big(y_t - f(\varphi_t, \hat{\theta})\Big) = 0, \tag{2.7}$$

is not desirable and an extreme case of *overfitting*. Overfitting is a problem that can occur when fitting a model and means that the model has been adjusted to the particular measurement noise realization. Overfitting is primarily a problem for flexible models and to chose a model structure just flexible enough to imitate f_0 (and not flexible enough to be able to imitate the noise) would be ideal.

There are a number of approaches to find what is "just flexible enough". Most approaches can be seen belonging to either *cross validation* or *regularization*.

2.5 Cross Validation

In *Cross Validation* (CV) the validation data set $\{(\varphi_t, y_t)\}_{t \in \mathcal{N}_v}$ is utilized to find what is "just flexible enough". Since the measurement noise e of the validation data set is impossible to predict, the best possible would be to perfectly predict the outcome of the deterministic part of (2.3) *i.e.,* $f_0(\varphi)$. Therefore, for a number of candidate models $f_i(\varphi, \hat{\theta}_i)$, $i = 1, \dots, m$ ($\hat{\theta}$ found using (2.6)), a model is chosen

by

$$\underset{f_i(\varphi,\hat{\theta}_i),i=1,\ldots,m}{\arg\min} \sum_{t\in\mathcal{N}_{\mathrm{v}}} l\Big(y_t - f_i(\varphi_t,\hat{\theta}_i)\Big). \tag{2.8}$$

This type of cross-validation is the most common in system identification. There are however several other types of cross validation, see *e.g.,* (Hastie et al., 2001, pp. 214-217).

To evaluate (2.8) we need to evaluate $f(\varphi,\theta)$ at the regressors of the validation data set. To compute predictions for f_0 at regressors not included in the estimation data set is called *generalization* (Bishop, 2006, p. 2). For most practical purposes it is not enough to find a model $f(\varphi,\hat{\theta})$ that well imitates f_0 at the estimation data set, generalization is therefore an important property of a model. This is sometimes referred to as the model's ability to *generalize* to unseen data.

2.6 Regularization

Regularization is in general a methodology for making an *ill-posed* problem *well-posed*, but regularization can also be used to control for overfit. We care for both these applications in this thesis. We however choose to focus on the type of regularization (referred to as a *standard regularization method* in Poggio et al. (1985)) obtained by adding a penalty term J to the criterion of fit. The penalty J should be regarded as a means to introduce *a priori* knowledge.

In particular, given a number of candidate models $f_i(\varphi,\hat{\theta}_i)$, $i = 1,\ldots,m$ ($\hat{\theta}$ found using (2.6)), we can use regularization to select a model "just flexible enough" by considering a criterion

$$\underset{f_i(\varphi,\hat{\theta}_i),i=1,\ldots,m}{\arg\min} \sum_{t\in\mathcal{N}_{\mathrm{e}}} l\Big(y_t - f_i(\varphi_t,\hat{\theta}_i)\Big) + J(f_i). \tag{2.9}$$

J should then be a flexibility penalty conveying the message that we wish an as "simple" model as possible that fits the estimation data. Notice that to choose a model using (2.9) only requires the estimation data set while cross-validation requires both an estimation and a validation data set. Regularization may therefore be a good choice when the number of observation data is limited.

The *Akaike Information Criterion* (AIC, Akaike (1973)),

$$\underset{f_i(\varphi,\hat{\theta}_i),i=1,\ldots,m}{\arg\min} \; -2\sum_{t\in\mathcal{N}_{\mathrm{e}}} \log p_e\Big(y_t - f_i(\varphi_t,\hat{\theta}_i)\Big) + 2dim(\hat{\theta}_i), \tag{2.10}$$

with $\hat{\theta}_i$ found using $l(\cdot) = -\log p_e(\cdot)$ in (2.6) (MLE of θ), is an example of this type of usage of regularization.

Example 2.1: ARX and Model Selection

Consider a single-input single-output dynamic system with an input u_t and an output y_t. Let the candidate models be ARX models with different nb's (see (2.4) for ARX and nb). Let *e.g.,*

$$f_1(\varphi_t, \theta_1) = \varphi_t^T \theta_1, \quad na = 1,\ nb = 1, \tag{2.11a}$$

$$f_2(\varphi_t, \theta_2) = \varphi_t^T \theta_2, \quad na = 1,\ nb = 2, \tag{2.11b}$$

$$\vdots$$

$$f_m(\varphi_t, \theta_m) = \varphi_t^T \theta_m, \quad na = 1,\ nb = m, \tag{2.11c}$$

and compute θ_1, $\theta_2, \ldots, \theta_m$ using (2.6). The flexibility of an ARX model grows with nb, a suitable choice of penalty J in (2.9) could therefore be

$$J(f_i) = nb \text{ for } f_i \tag{2.12}$$

if an as "simple" model as possible but with a reasonable good fit is sought.

Regularization can also be used to control the regressor parameter value of a single model. $f(\varphi_t, \theta)$ is then adjusted to the observations by choosing θ as

$$\hat{\theta} = \arg\min_{\theta} \sum_{t \in \mathcal{N}_e} l\Big(y_t - f(\varphi_t, \theta)\Big) + \lambda J(\theta, \varphi_t), \tag{2.13}$$

rather than using (2.6). $J(\theta, \varphi_t)$ again serves as a cost on flexibility and is often used to penalize non-smooth estimates (this is discussed in Chapter 5). However, $J(\theta, \varphi_t)$ could also be used to express the prior knowledge of a sparse parameter vector θ (this is discussed in Chapter 4). $\lambda \in \mathcal{R}^+$ is seen as a design parameter and regulates the trade-off between fit to the estimation data and flexibility. Choosing the "just flexible enough" model structure is now a matter of choosing the right λ-value. λ is denoted the *regularization parameter* or *regularization constant* and $\hat{\theta}$ as a function of regularization parameter, the *regularization path*.

An expression of the form (2.13) is of great importance for this thesis and will be a key ingredient in the theory developed in Chapters 4 and 5 and in several of the papers of Part II. (2.13) is a type of *shrinkage method* as it is often used to shrink regression parameters toward zero (Hastie et al., 2001, p. 59).

Example 2.2: ARX and ℓ_2-Regularization

Consider again a single-input single-output dynamic system with an input u_t and an output y_t. Let us use an ARX model (2.4) and fix na and nb.

Let $l(\cdot) = (\cdot)^2$ in (2.6). For this particular choice, (2.6) is referred to as the *Least Squares* (LS) problem. Let $\{(\varphi_t, y_t)\}_{t=1}^{N_e}$ be a given estimation data set. If we now define

$$y \triangleq \begin{bmatrix} y_1 & \ldots & y_{N_e} \end{bmatrix}^T, \qquad \Phi \triangleq \begin{bmatrix} \varphi_1 & \ldots & \varphi_{N_e} \end{bmatrix}^T, \tag{2.14}$$

(2.6) can be written as

$$\hat{\theta} = \arg\min_{\theta} \|y - \Phi\theta\|_2^2 = \arg\min_{\theta} (y - \Phi\theta)^T (y - \Phi\theta). \tag{2.15}$$

We can characterize the solution of (2.15) by determining if

$$y = \Phi\theta \tag{2.16}$$

is *overdetermined*, *underdetermined* or has a unique solution. It is useful to separate between the three cases:

(2.16) **is overdetermined.** In this case there are more observations than model parameters. This is the most studied case in system identification. If Φ has full column rank *i.e.,*

$$rank(\Phi) = dim(\theta), \tag{2.17}$$

then $\hat{\theta}$ in (2.15) can be computed explicitly to

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T y. \tag{2.18}$$

$(\Phi^T\Phi)^{-1}\Phi^T$ is known as the *Moore-Penrose pseudoinverse* and generally denoted by $\Phi^\dagger$. Geometrically, $\Phi\theta$ is a linear combination of the columns of Φ. $f(\varphi, \theta)$ is hence restricted to the plane spanned by the columns of Φ. (2.15) can then be interpreted as the problem of finding the vector in the plane spanned by the columns of Φ that is the closest, in an Euclidean sense, to the vector y. The orthogonal projection of y onto the plane spanned by the columns of Φ,

$$\Phi(\Phi^T\Phi)^{-1}\Phi^T y, \tag{2.19}$$

is well known to minimize this distance. (2.18) should therefore be seen as a projection onto the plane spanned by the columns of Φ. When Φ has full rank, (2.15) has a unique solution. If Φ does not have full rank, there exits a lower number columns ($< dim(\theta)$) that span the plane. $\hat{\theta}$ is therefore no longer unique.

The ARX model $f(\varphi_t, \hat{\theta})$ does not, in general, perfectly predict the outputs in the estimation data set, but since measurement noise is present, this is preferred over an overfit.

(2.16) **has a unique solution.** Assume Φ is quadratic and has full rank, $\mathcal{R}^{N_e}$ is then spanned by the columns of Φ which also make up a basis for $\mathcal{R}^{N_e}$. The task is now to express y in this basis. We hence want to solve the equation system

$$y = \Phi\theta. \tag{2.20}$$

(2.20) is solved by

$$\hat{\theta} = \Phi^{-1} y. \tag{2.21}$$

The inverse exists since Φ is quadratic and has full rank. For $\hat{\theta} = \Phi^{-1} y$ a

perfect fit is obtained, *i.e.*,

$$\|y - \Phi\hat{\theta}\|_2^2 = 0. \tag{2.22}$$

It is worth notice that the Moore-Penrose pseudoinverse in this case reduces to the ordinary inverse since

$$\Phi^\dagger = (\Phi^T\Phi)^{-1}\Phi^T = \Phi^{-1}\Phi^{-T}\Phi^T = \Phi^{-1}. \tag{2.23}$$

(2.18) hence still holds.

(2.16) **is underdetermined.** In this case, the columns of Φ construct an over complete basis for $\mathcal{R}^{N_e}$. There is hence an infinite number of θs that obtain a perfect fit *i.e.*,

$$\|y - \Phi\theta\|_2^2 = 0. \tag{2.24}$$

(2.15) is hence ill-posed. Regularization can here be used to express which one of these infinite solutions that is desired.

The Moore-Penrose pseudoinverse is for this case not well defined, since $\Phi^T\Phi$ is singular.

Remark 2.1. If (2.16) is either overdetermined or has a unique solution, (2.15) is a *strictly convex* optimization problem and has therefore a unique solution (see *e.g.*, Bertsekas et al. (2003, Prop. 2.1.2)). If (2.16) is underdetermined, (2.15) is *convex* and any minimizing $\hat{\theta}$ is therefore a *global minimum* (see *e.g.*, Boyd and Vandenberghe (2004, p. 138)). $\hat{\theta}$ may however not be unique in this case.

Let us assume that we have insight that tells us that θ should be "small". We could then use regularization to reduce the flexibility of $f(\varphi, \theta) = \varphi^T\theta$ and to only allow models with a small θ. That would *e.g.*, help us find a unique model if (2.16) is underdetermined. However, it could also be used to reduce the flexibility of a model to control for overfit and find a "just flexible enough" model ((2.16) does not need to be underdetermined to use regularization for this purpose). Let us say that we would be satisfied if $\|\theta\|_2^2$ is kept small. Using regularization we can express this prior knowledge/insight as

$$\hat{\theta} = \arg\min_\theta \|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_2^2, \quad \lambda \in \mathcal{R}^+. \tag{2.25}$$

(2.25) is an ℓ_2*-regularized least squares problem*, often referred to as *ridge regression* or *Tikhonov regularization* (Hoerl and Kennard (1970), see also Hastie et al. (2001), p. 59). Since the objective function is quadratic in θ, an explicit expression for $\hat{\theta}$ can be computed. The gradient with respect to θ of the objective function of (2.25) becomes

$$\nabla_\theta \left(\|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_2^2 \right) = -2\Phi^T(y - \Phi\theta) + 2\lambda\theta. \tag{2.26}$$

Setting the gradient equal to zero and solve gives

$$\hat{\theta} = (\Phi^T\Phi + \lambda I)^{-1}\Phi^T y. \tag{2.27}$$

(2.27) and (2.18) take a very similar form. And in fact, adding a small diagonal

matrix λI to $\Phi^T\Phi$ to make the Moore-Penrose pseudoinverse well defined was the main motivation for ridge regression when it was introduced by Hoerl and Kennard (1970).

2.7 Bias-Variance Tradeoff

Let us assume that an estimate of f_0 at the regressor φ_* is desired. To find what is "just flexible enough" can then be shown to be a matter of finding a suitable tradeoff between *variance*

$$\mathsf{E}_{\hat{\theta}}\left[\left(\mathsf{E}_{\hat{\theta}}[f(\varphi_*,\hat{\theta})]-f(\varphi_*,\hat{\theta})\right)^2\right] \tag{2.28}$$

and *bias*

$$f_0(\varphi_*)-\mathsf{E}_{\hat{\theta}}[f(\varphi_*,\hat{\theta})]. \tag{2.29}$$

This can be understood as follows. Given an estimation data set, we estimate θ. Since the y-measurements in the estimation data set are noisy, they are inherently stochastic and so will also $\hat{\theta}$ be. It therefore makes sense to study the quantity

$$\mathsf{E}_{\hat{\theta}}\left[\left(f_0(\varphi_*)-f(\varphi_*,\hat{\theta})\right)^2\right] \tag{2.30}$$

as a measure of performance (for estimating f_0 at φ_*). The expectation is here taken with respect to $\hat{\theta}$. This quantity is called the *Mean Squared Error* (MSE). To minimize the MSE would be ideal and was earlier referred to as finding a model "just flexible enough". To see how the bias and variance relate to MSE, add and subtract $\mathsf{E}_{\hat{\theta}}[f(\varphi_*,\hat{\theta})]$ in (2.30). We get

$$\begin{aligned}
\mathsf{E}_{\hat{\theta}}\left[\left(f_0(\varphi_*)-f(\varphi_*,\hat{\theta})\right)^2\right] &= \mathsf{E}_{\hat{\theta}}\left[\left(f_0(\varphi_*)-\mathsf{E}_{\hat{\theta}}[f(\varphi_*,\hat{\theta})]+\mathsf{E}_{\hat{\theta}}[f(\varphi_*,\hat{\theta})]-f(\varphi_*,\hat{\theta})\right)^2\right] \\
&= \mathsf{E}_{\hat{\theta}}\Big[\left(f_0(\varphi_*)-\mathsf{E}_{\hat{\theta}}[f(\varphi_*,\hat{\theta})]\right)^2+\left(\mathsf{E}_{\hat{\theta}}[f(\varphi_*,\hat{\theta})]-f(\varphi_*,\hat{\theta})\right)^2 \\
&\quad +2\left(f_0(\varphi_*)-\mathsf{E}_{\hat{\theta}}[f(\varphi_*,\hat{\theta})]\right)\left(\mathsf{E}_{\hat{\theta}}[f(\varphi_*,\hat{\theta})]-f(\varphi_*,\hat{\theta})\right)\Big] \\
&= \left(f_0(\varphi_*)-\mathsf{E}_{\hat{\theta}}[f(\varphi_*,\hat{\theta})]\right)^2+\mathsf{E}_{\hat{\theta}}\left[\left(\mathsf{E}_{\hat{\theta}}[f(\varphi_*,\hat{\theta})]-f(\varphi_*,\hat{\theta})\right)^2\right].
\end{aligned}$$

The first term

$$\left(f_0(\varphi_*)-\mathsf{E}_{\hat{\theta}}[f(\varphi_*,\hat{\theta})]\right)^2 \tag{2.31}$$

is the squared bias and the second term

$$\mathsf{E}_{\hat{\theta}}\left[\left(\mathsf{E}_{\hat{\theta}}[f(\varphi_*,\hat{\theta})]-f(\varphi_*,\hat{\theta})\right)^2\right] \tag{2.32}$$

is the variance. The bias is due to limitations in our model structure and the variance term is due to the stochastic nature of our estimation data set (the measurement noise). However, both the bias and the variance also depend on the cost function used to find $\hat{\theta}$.

Flexible models generally give high variance, but low bias, whereas non-flexible models give low variance, but high bias.

Example 2.3: Regularization and the Bias-Variance Tradeoff

Consider the single-input single-output system ($\delta(\cdot)$ the Dirac delta function)

$$y_t = \sum_{k=1}^{n} g_k^0 u_{t-k} + e_t, \quad \mathsf{E}[e_t] = 0,\ \mathsf{E}[e_t e_s] = \delta(t-s)\sigma^2,\ \forall t, s \in \mathcal{N}. \tag{2.33}$$

The sequence $\{g_k^0\}_{k=1}^n$ is the *impulse response* of the system *i.e.,* the response to an impulse ($u_t = \delta(t)$ in (2.33) gives $y_t = g_t^0 + e_t$, $t = 1, \dots, n$, $y_t = e_t$, $t = n+1, n+2, \dots$). Let us estimate the impulse response. Assume that we use an nth order FIR model (see (2.5))

$$f(\varphi_t, \theta) = \varphi_t^T \theta, \quad \varphi_t = \begin{bmatrix} u(t-1) & \dots & u(t-n) \end{bmatrix}^T,\ \theta \in \mathcal{R}^n. \tag{2.34}$$

Let $\{(\varphi_t, y_t)\}_{t=1}^{N_\mathrm{e}}$ be the estimation data set and define

$$\begin{aligned} y &\triangleq \begin{bmatrix} y_1 & \dots & y_{N_\mathrm{e}} \end{bmatrix}^T, \quad \Phi \triangleq \begin{bmatrix} \varphi_1 & \dots & \varphi_{N_\mathrm{e}} \end{bmatrix}^T, \\ \Lambda &\triangleq \begin{bmatrix} e_1 & \dots & e_{N_\mathrm{e}} \end{bmatrix}^T, \quad \theta_0 \triangleq \begin{bmatrix} g_1^0 & \dots & g_n^0 \end{bmatrix}^T. \end{aligned} \tag{2.35}$$

Consider now the ℓ_2-regularized least squares criterion

$$\hat{\theta} = \arg\min_{\theta} \|y - \Phi\theta\|_2^2 + \theta^T D \theta, \quad D \in \mathcal{R}^{n \times n},\ D \geq 0, \tag{2.36}$$

with a solution (see (2.27))

$$\hat{\theta} = (\Phi^T\Phi + D)^{-1}\Phi^T y. \tag{2.37}$$

The bias for an estimate at φ_* is then readily computed to

$$\begin{aligned} \varphi_*^T\theta_0 - \mathsf{E}_{\hat{\theta}}[\varphi_*^T\hat{\theta}] &= \varphi_*^T\theta_0 - \mathsf{E}_y[\varphi_*^T(\Phi^T\Phi + D)^{-1}\Phi^T y] && \text{(2.38a)} \\ &= \varphi_*^T\theta_0 - \varphi_*^T(\Phi^T\Phi + D)^{-1}\Phi^T\, \mathsf{E}_\Lambda[\Phi\theta_0 + \Lambda] && \text{(2.38b)} \\ &= \varphi_*^T\theta_0 - \varphi_*^T(\Phi^T\Phi + D)^{-1}\Phi^T\Phi\theta_0 && \text{(2.38c)} \end{aligned}$$

and the variance to

$$\begin{aligned} \mathsf{E}_{\hat{\theta}}\left[\left(\mathsf{E}_{\hat{\theta}}[\varphi_*^T\hat{\theta}] - \varphi_*^T\hat{\theta}\right)^2\right] &= \mathsf{E}_y\left[\left(\varphi_*^T(\Phi^T\Phi + D)^{-1}\Phi^T\Phi\theta_0 - \varphi_*^T(\Phi^T\Phi + D)^{-1}\Phi^T y\right)^2\right] \\ &= \mathsf{E}_\Lambda\left[\left(\varphi_*^T(\Phi^T\Phi + D)^{-1}\Phi^T\left(\Phi\theta_0 - (\Phi\theta_0 + \Lambda)\right)\right)^2\right] \\ &= \mathsf{E}_\Lambda\left[\left(\varphi_*^T(\Phi^T\Phi + D)^{-1}\Phi^T\Lambda\right)^2\right] \\ &= \varphi_*^T(\Phi^T\Phi + D)^{-1}\Phi^T\, \mathsf{E}_\Lambda[\Lambda\Lambda^T]\Phi(\Phi^T\Phi + D)^{-1}\varphi_* \\ &= \sigma^2\varphi_*^T(\Phi^T\Phi + D)^{-1}\Phi^T\Phi(\Phi^T\Phi + D)^{-1}\varphi_*. \end{aligned} \tag{2.39}$$

Let now $D = \lambda I_n$, $\lambda \geq 0$. Then, if the estimation data input u_t is chosen as zero

mean white noise with variance μ and for a large N_e, it holds that

$$\Phi^T \Phi \approx N_e \mu I_n. \tag{2.40}$$

If $\Phi^T \Phi = N_e \mu I_n$ is used in (2.38) and (2.39) the bias becomes

$$\varphi_*^T \theta_0 - \mathsf{E}_{\hat{\theta}}[\varphi_*^T \hat{\theta}] = \left(\frac{\lambda}{N_e \mu + \lambda} \right) \varphi_*^T \theta_0 \tag{2.41}$$

and the variance

$$\mathsf{E}_{\hat{\theta}} \left[\left(\mathsf{E}_{\hat{\theta}}[\varphi_*^T \hat{\theta}] - \varphi_*^T \hat{\theta} \right)^2 \right] = \sigma^2 \frac{N_e \mu}{(N_e \mu + \lambda)^2} \varphi_*^T \varphi_*. \tag{2.42}$$

Notice that when $\lambda = 0$ we obtain the unbiased least squares estimate. The variance for the least squares estimate is however larger than the variance of an estimate obtained for a small positive λ. A small positive λ causes a biased estimate though. Figure 2.1 gives a sketch of how the typical variance and bias depend on λ.

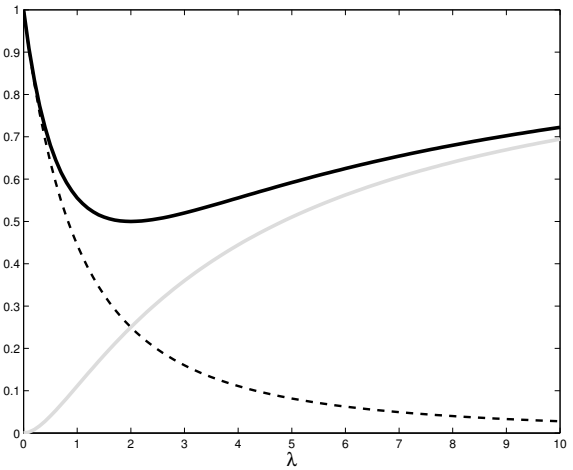

Figure 2.1: *Bias-variance visualization for regularization. The squared bias* $\left(f_0(\varphi_*) - \mathsf{E}_{\hat{\theta}}[f(\varphi_*, \hat{\theta})] \right)^2$ *is showed using the gray line, the variance* $\mathsf{E}_{\hat{\theta}} \left[\left(\mathsf{E}_{\hat{\theta}}[f(\varphi_*, \hat{\theta})] - f(\varphi_*, \hat{\theta}) \right)^2 \right]$ *using the dashed line and the MSE using the black line.*

We will return to impulse response identification in Paper F and explore more sophisticated choices of D-matrix. In fact, some of the most recent contributions in impulse response identification use ℓ_2-regularization, see *e.g.,* Pillonetto and De Nicolao (2010).

2.8 Performance Measures

To evaluate the prediction performance of different models a performance measure is needed. For a given test data set $\{(\varphi_t, y_t)\}_{t\in\mathcal{N}_\text{t}}$ and a model $f(\varphi, \hat{\theta})$, we choose to use

$$\left(1 - \sqrt{\frac{\sum_{t\in\mathcal{N}_\text{t}} \left|y_t - f(\varphi_t, \hat{\theta})\right|^2}{\sum_{t\in\mathcal{N}_\text{t}} \left|y_t - \frac{1}{N_\text{t}} \sum_{s\in\mathcal{N}_t} y_s\right|^2}}\right) \times 100 \tag{2.43}$$

as a performance measure. We will call the computed quantity *fit* and express us by saying that a prediction has a certain percentage fit to a set of data.

At some point in the thesis the *Mean Absolute Error* (MAE)

$$\frac{1}{N_\text{t}} \sum_{t\in\mathcal{N}_\text{t}} \left|y_t - f(\varphi_t, \hat{\theta})\right| \tag{2.44}$$

will also be used.

2.9 Bayesian Modeling

In *Bayesian modeling*, or *Bayesian inference*, probability distributions are used to represent stochasticity and uncertainty. For a parametric model, this implies that a distribution over parameter-values is computed rather than a single regressor parameter estimate $\hat{\theta}$. Also the predictions will be distributions over possible estimates rather than a single function-value for a given φ.

A Bayesian practitioner argues that there are two sources of information. The prior knowledge about the system and the observations. The prior knowledge or prior believes have to be formulated as a probability distribution, denoted a *prior*. The prior believes then get updated using observations to form a posterior, an updated probability distribution. How to weight together the prior and the observations is given by Bayes' theorem (Bayes, 1763):

Theorem 2.1 (Bayes' Theorem). *Let $p(\theta)$ be a prior, $p\left(\{y_t\}_{t\in\mathcal{N}_\text{e}} \middle| \theta, \{\varphi_t\}_{t\in\mathcal{N}_\text{e}}\right)$ the likelihood of observing the outputs $\{y_t\}_{t\in\mathcal{N}_\text{e}}$ given $\{\varphi_t\}_{t\in\mathcal{N}_\text{e}}$ and θ, and $p(\{y_t\}_{t\in\mathcal{N}_\text{e}}|\{\varphi_t\}_{t\in\mathcal{N}_\text{e}})$ the probability of observing the data $\{y_t\}_{t\in\mathcal{N}_\text{e}}$ given $\{\varphi_t\}_{t\in\mathcal{N}_\text{e}}$. The posterior distribution for θ given the observations is then given by*

$$p\left(\theta \middle| \{(\varphi_t, y_t)\}_{t\in\mathcal{N}_\text{e}}\right) = \frac{p\left(\{y_t\}_{t\in\mathcal{N}_\text{e}} \middle| \theta, \{\varphi_t\}_{t\in\mathcal{N}_\text{e}}\right) p(\theta)}{p\left(\{y_t\}_{t\in\mathcal{N}_\text{e}} \middle| \{\varphi_t\}_{t\in\mathcal{N}_\text{e}}\right)}. \tag{2.45}$$

The model $f(\varphi, \theta)$ is in a Bayesian framework represented by the *predictive distribution*. Let y_* be an observation of $f_0(\varphi_*)$, $p\left(\theta \middle| \{(\varphi_t, y_t)\}_{t\in\mathcal{N}_\text{e}}\right)$ the posterior distribution for θ given the observations (computed using Theorem 2.1) and let $p(y_*|\varphi_*, \theta)$ be the likelihood of observing the output y_* given φ_* and θ. The pre-

dictive distribution for y_* is then given by

$$p\Big(y_*\Big|\{(\varphi_t, y_t)\}_{t\in\mathcal{N}_\mathrm{e}}, \varphi_*\Big) = \int p\Big(y_*\Big|\varphi_*, \theta\Big)p\Big(\theta\Big|\{(\varphi_t, y_t)\}_{t\in\mathcal{N}_\mathrm{e}}\Big)d\theta. \tag{2.46}$$

The predictive distribution tells us how certain we are that the measured system response to φ_* takes a certain value.

It is common to let the prior $p(\theta)$ depend on a number of *hyperparameters*, let us call these θ_h. The prior hence takes the form $p(\theta|\theta_h)$. The hyperparameters are usually determined from data by maximizing the log marginal likelihood,

$$\log p\Big(\{y_t\}_{t\in\mathcal{N}_\mathrm{e}}\Big|\{\varphi_t\}_{t\in\mathcal{N}_\mathrm{e}}, \theta_h\Big) = \log \int p\Big(\{y_t\}_{t\in\mathcal{N}_\mathrm{e}}\Big|\{\varphi_t\}_{t\in\mathcal{N}_\mathrm{e}}, \theta\Big)p(\theta|\theta_h)d\theta. \tag{2.47}$$

This approach to estimating θ_h is referred to as *empirical Bayes* (see *e.g.,* Bishop (2006, p. 165)).

Example 2.4: ARX Cont'd

Consider the ARX-type of system

$$y_t = \varphi_t^T\theta + e_t, \quad e_t \sim N(0, \sigma^2), \tag{2.48}$$

with φ_t containing old system inputs and outputs. Assume that we are given the observations $\{(\varphi_t, y_t)\}_{t=1}^{N_\mathrm{e}}$, know the (i.i.d.) measurement noise variance σ^2 and that we have reason to believe that θ is small. Taking a Bayesian approach, we then compute the posterior distribution $p\Big(\theta\Big|\{(\varphi_t, y_t)\}_{t=1}^{N_\mathrm{e}}\Big)$ as (see Theorem 2.1)

$$p\Big(\theta\Big|\{(\varphi_t, y_t)\}_{t=1}^{N_\mathrm{e}}\Big) = \frac{\Pi_{t=1}^{N_\mathrm{e}} N(y_t; \varphi_t^T\theta, \sigma^2)p(\theta)}{p\Big(\{y_t\}_{t=1}^{N_\mathrm{e}}\Big|\{\varphi_t\}_{t=1}^{N_\mathrm{e}}\Big)}. \tag{2.49}$$

$p(\theta)$ is here the prior and $N(y_t; \varphi_t^T\theta, \sigma^2)$ is used to denote that $y_t \sim N(\varphi_t^T\theta, \sigma^2)$. To convey our belief of a small θ, and to get a closed-form expression for the posterior, we choose to use a Gaussian prior, say $N(0, I)$. If we first introduce

$$y \triangleq \begin{bmatrix} y_1 & \dots & y_{N_\mathrm{e}} \end{bmatrix}^T, \ \Phi \triangleq \begin{bmatrix} \varphi_1 & \dots & \varphi_{N_\mathrm{e}} \end{bmatrix}^T, \tag{2.50}$$

the posterior can be computed using standard Gaussian identities, see *e.g.,* Rasmussen and Williams (2005, p. 200), to

$$p\Big(\theta\Big|\{(\varphi_t, y_t)\}_{t=1}^{N_\mathrm{e}}\Big) = \frac{\Pi_{t=1}^{N_\mathrm{e}} N(y_t; \varphi_t^T\theta, \sigma^2)N(\theta; 0, I)}{\int \Pi_{t=1}^{N_\mathrm{e}} N(y_t; \varphi_t^T\theta, \sigma^2)N(\theta; 0, I)d\theta} \tag{2.51a}$$

$$= N(\theta; (\Phi^T\Phi + \sigma^2 I)^{-1}\Phi^T y, (\sigma^{-2}\Phi^T\Phi + I)^{-1}). \tag{2.51b}$$

The predictive distribution is now readily computed to

$$p\Big(y_*\Big|\varphi_*, \{(\varphi_t, y_t)\}_{t=1}^{N_\mathrm{e}}\Big) = \int N(y_*; \varphi_*^T\theta, \sigma^2)p\Big(\theta\Big|\{(\varphi_t, y_t)\}_{t=1}^{N_\mathrm{e}}\Big)d\theta \tag{2.52}$$

$$= N\Big(\varphi_*^T(\Phi^T\Phi + \sigma^2 I)^{-1}\Phi^T y, \sigma^2 + \varphi_*^T(\sigma^{-2}\Phi^T\Phi + I)^{-1}\varphi_*\Big),$$

with $p\left(\theta \middle| \{(\varphi_t, y_t)\}_{t=1}^{N_e}\right)$ from (2.51).

Let us now explore what happens if we let the variance of the prior free and instead uses $N(0, \theta_h I)$, $\theta_h \in \mathcal{R}^+$, as a prior. We then see θ_h as a hyperparameter and compute it by maximizing the log marginal likelihood. Using basic Gaussian identities (see *e.g.*, Rasmussen and Williams (2005, p. 200)), (2.47) can in this particular setting be expressed as

$$\log p\left(\{y_t\}_{t=1}^{N_e} \middle| \{\varphi_t\}_{t=1}^{N_e}, \theta_h\right) = \log \int \Pi_{t=1}^{N_e} N(y_t; \varphi_t^T \theta, \sigma^2) N(\theta; 0, \theta_h I) d\theta \tag{2.53a}$$

$$= \log Z^{-1} \int N\left(\theta; \sigma^{-2} A^{-1} \Phi^T y, A^{-1}\right) d\theta \tag{2.53b}$$

$$= \log Z^{-1} \tag{2.53c}$$

with A and the normalizing constant Z defined as

$$Z^{-1} \triangleq \frac{1}{\theta_h^{dim(\theta)/2}} \frac{1}{(2\pi\sigma^2)^{N_e/2}} |A|^{-1/2} e^{-\frac{1}{2\sigma^2}\|y - \sigma^{-2}\Phi A^{-1}\Phi^T y\|_2^2 - \frac{1}{2\theta_h \sigma^4}\|A^{-1}\Phi^T y\|_2^2} \tag{2.54}$$

$$A \triangleq \theta_h^{-1} I + \sigma^{-2} \Phi^T \Phi. \tag{2.55}$$

θ_h is then chosen according to

$$\hat{\theta}_h = \arg\max_{\theta_h} \log Z^{-1}. \tag{2.56}$$

For more details see *e.g.*, Bishop (2006, pp. 152-158 and pp. 165-169).

Remark 2.2. Maximizing the posterior $p\left(\theta \middle| \{(\varphi_t, y_t)\}_{t \in \mathcal{N}_e}\right)$ with respect to θ gives the *Maximum A Posteriori* (MAP) estimate for θ. When the posterior is a Gaussian, the MAP is given by the mean of the Gaussian. In Example 2.4, using $N(0, I)$ as a prior, the MAP estimate for θ became

$$(\Phi^T \Phi + \sigma^2 I)^{-1} \Phi^T y. \tag{2.57}$$

This is the same expression as for ridge regression with $\lambda = \sigma^2$, see (2.27). In fact, most standard regularization methods can be given an interpretation as a MAP estimate.

2.10 High Dimensional Regression and Manifolds

We finish this chapter on mathematical modeling and regression by discussing high dimensional regression, manifolds and manifold learning. We will return to these subjects in Paper E.

High-dimensional regressors can lead to ill-posed regression problems. Especially if the dimension of the regressors exceeds the number of observations, special care is needed, as we saw in Example 2.2. There are a number of strategies for handling high-dimensional regression problems:

- The first strategy is *feature selection*. Feature selection is used to reduce the dimension of the high-dimensional regressors by eliminating elements

having *e.g.,* little correlation with the output. The "new" low-dimensional regressors are used instead of the original regressors in the regression algorithm. An example is *backward stepwise regression* (see *e.g.,* Daniel and Wood (1980, pp. 84-85)). Also many regression methods using regularization contain some type of feature selection. Popular regression methods here include lasso (see *e.g.,* Example 4.1) and ridge regression.

- The second strategy is *feature extraction*. Feature extraction is also used to reduce the dimension of the high-dimensional regressors. However, rather than eliminating elements, elements are combined. *Partial Least Squares* (PLS, Wold (1966)) and *Principle Component Analysis* (PCA, Pearson (1901)) are popular methods used for feature extraction. Also *manifold learning* discussed in the next section can be used for feature extraction. The regression method discussed in Paper E can also be seen using feature extraction.

Both feature selection and extraction are special cases of *dimensionality reduction* methods.

Another issue which high-dimensional regression algorithms have to deal with is the lack of data, commonly termed the *curse of dimensionality* (Bellman, 1961). For instance, imagine N samples uniformly distributed in a d-dimensional unit hypercube $[0, 1]^d$. The N samples could for example be the regressors in the set of observed data. To include 10% of the samples, we need on average to pick out a cube with the side 0.1 for $d = 1$ and a cube with the side 0.8 for $d = 10$, Figure 2.2 illustrates this. The data hence easily become sparse with increasing

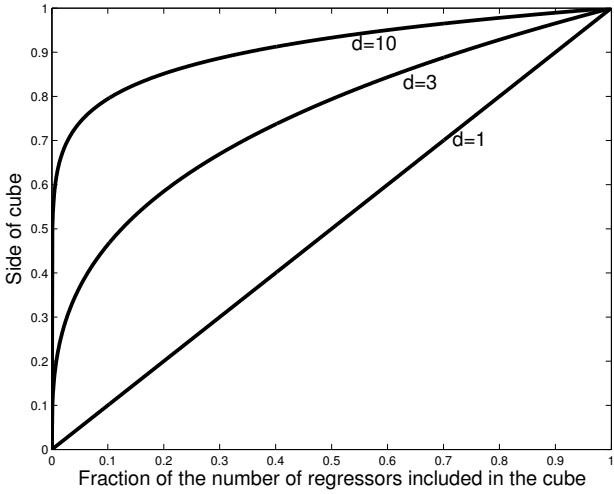

Figure 2.2: *An illustration of the curse of dimensionality. Assume that the N regressors are uniformly distributed in a d-dimensional unit cube. On average we then need to use a cube with a side of 0.1 to include $0.1N$ regressors for $d = 1$, while for $d = 10$ we will need a cube with a side of 0.8.*

dimensionality. Consequently, given a regressor, the likelihood of finding one of the estimation regressors close-by, gets smaller and smaller with increasing dimension. This means that for high-dimensional regression problems, considerably more samples are needed than for low-dimensional regression problems to make accurate predictions. This also implies that regression methods using pairwise distances between regressors, such as *nearest neighbor* (see *e.g.,* Hastie et al. (2001, p. 14)) and *support vector regression* (see Section 5.1), suffer. This follows since, as dimensionality grows the distances between regressors increase, become more similar and hence less expressive (see Figure 2.3 for an illustration and Chapelle et al. (2006) and Bengio et al. (2006) for further readings).

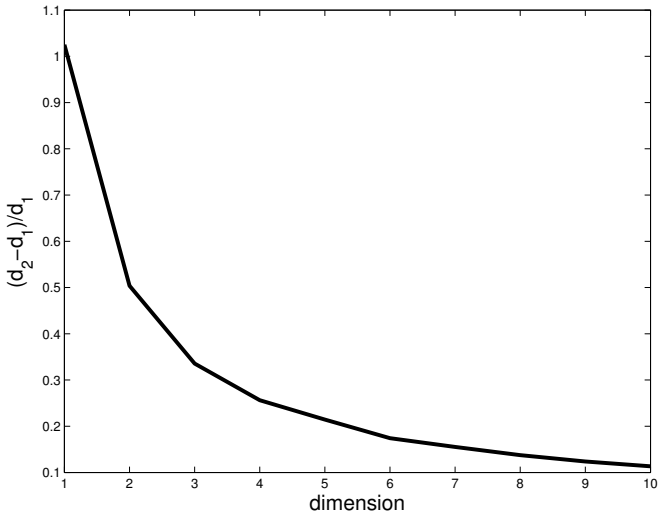

Figure 2.3: *As the dimension of the regressor space increases (keeping the number of regressors fixed) so does the distance from any regressor to all other regressors. The distance to the closest estimation regressor, d_1, of a regressor is hence increasing with dimension. The distance to the second closest estimation regressor, d_2, is also increasing. A prediction has then to be made based on more and more distant observations. In addition, the relative distance, $(d_2 - d_1)/d_1$, decreases, making the estimation data less expressive. Rephrased in a somewhat sloppy way, a given point in a high-dimensional space has many "nearest neighbors", but all far away.*

Very common, however, is that the regressors $\varphi \in \mathcal{R}^{n_\varphi}$ for various reasons are constrained to lie in a subset $\Omega \subset \mathcal{R}^{n_\varphi}$. A specific example could be a set of images of human faces. An image of a human face is a $p \times p$ matrix, each entry of the matrix giving the gray tone in a pixel. If we vectorize the image, the image becomes a point in $\mathcal{R}^{p^2}$. However, since features, such as eyes, mouth and nose, will be found in all images, the images will not be uniformly distributed in $\mathcal{R}^{p^2}$.

It is of special interest if Ω is a manifold.

Definition 2.1 (Manifold). A space $\mathcal{M} \subseteq \mathcal{R}^{n_\varphi}$ is said to be a n_z-dimensional *manifold* if there for every point $\varphi \in \mathcal{M}$ exists an open set $\mathcal{O} \subseteq \mathcal{M}$ satisfying:

- $\varphi \in \mathcal{O}$.
- $\mathcal{O}$ is *homeomorphic* to $\mathcal{R}^{n_z}$, meaning that there exists a one-to-one relation between $\mathcal{O}$ and a set in $\mathcal{R}^{n_z}$.

For details see *e.g.,* Lee (2000, p. 33).

For the set of $p \times p$ pixel images of human faces *e.g.,* the constraints implied by the different features characterizing a human face, make the images reside on a manifold enclosed in $\mathcal{R}^{p^2}$, see *e.g.,* Zhang et al. (2004). For *fMRI* (functional Magnetic Resonance Imaging) the situation is similar. For further discussions on fMRI data and manifolds, see Shen and Meyer (2005); Thirion and Faugeras (2004); Hu et al. (2006). Basically all sets of data for which data points can be parameterized using a set of parameters (fewer than the number of dimensions of the data) reside on a manifold. Any algebraic relation between regressor elements will therefore lead to regressors constrained to a manifold.

It is convenient to introduce the term *intrinsic description* for a n_z-dimensional parameterization of a manifold $\mathcal{M}$. We will not associate any properties to this description more than that it is n_z-dimensional. An intrinsic description of a one-dimensional manifold could for example be the distance from a specific point.

Remark 2.3. To express regressors in an intrinsic description is a way of doing feature extraction. Using an intrinsic description of the regressors instead of the original regressors in the regression algorithm may therefore be a way of making the regression problem well-posed, see *e.g.,* Ohlsson et al. (2007).

We illustrate the concepts of a manifold and intrinsic description with an example.

Example 2.5: Manifold and Intrinsic Description

Lines and circles are examples of one-dimensional manifolds. A two-dimensional manifold could for example be the surface of the earth. An intrinsic description associated with a manifold is a parametrization of the manifold, for example latitude and longitude for the earth surface manifold. Since the *Universal Transverse Mercator* (UTM) coordinate system is another two-dimensional parametrization of the surface of the earth and an intrinsic description, an intrinsic description is not unique.

A common assumption in regression is to assume smoothness. We will refer to the following assumption as the smoothness assumption:

Assumption A1 (The Smoothness Assumption). If two regressors φ_1, φ_2 are close, then so should their corresponding outputs $f_0(\varphi_1)$, $f_0(\varphi_2)$ be.

If regressors are constrained to a manifold there is an alternative to the smoothness assumption, commonly referred to as semi-supervised smoothness assumption. The semi-supervised smoothness assumption reads (Chapelle et al., 2006):

Assumption A2 (The Semi-Supervised Smoothness Assumption). Two outputs $f_0(\varphi_1)$, $f_0(\varphi_2)$ are assumed close if their corresponding regressors φ_1, φ_2 are close on the manifold.

"Close on the manifold" here means that there is a short path included in the manifold between the two regressors. The concept of geodesic distance is here useful. The *geodesic distance* between two points on a manifold $\mathcal{M}$ is the length of the shortest path included in $\mathcal{M}$ between the two points. The geodesic distance is assumed to be measured in the metric of the space in which the manifold is embedded. "Close on the manifold" can therefore be replaced by "close in terms of geodesic distance".

It should be noticed that the semi-supervised smoothness assumption is less conservative than the smoothness assumption. Hence, a function satisfying the semi-supervised smoothness assumption does not necessarily need to satisfy the smoothness assumption. Assumption A2 is illustrated in Example 2.6.

Example 2.6: The Semi-Supervised Smoothness Assumption

Assume that we are given a set of output-regressor pairs as shown in Figure 2.4. The regressors contain the position data (latitude, longitude) of an airplane

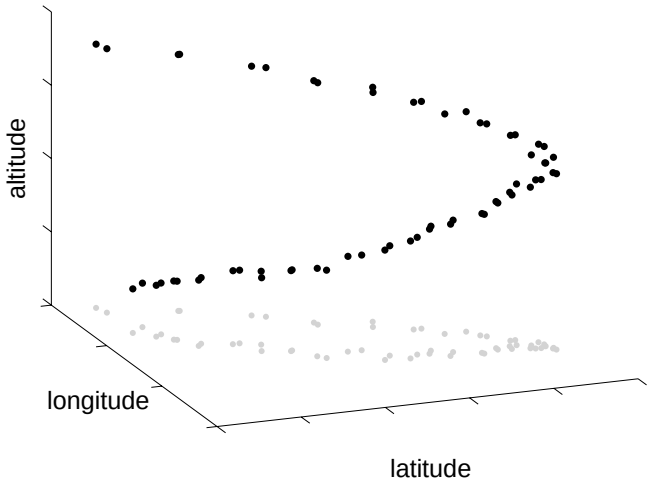

Figure 2.4: *Longitude, latitude and altitude measurement (black dots) of an airplane shortly after takeoff. Gray dots show the black dots projection onto the regressor space.*

shortly after takeoff. The output is chosen as the altitude of the airplane. The regressors thus being in $\mathcal{R}^2$ and the regressor/output space is $\mathcal{R}^3$. After takeoff the plane makes a turn during climbing and more or less returns along the same path in latitude and longitude as it just flown. The flight path becomes a one-dimensional curve, a manifold, in $\mathcal{R}^3$. However, the regressors for this path also belong to a curve, a manifold, in $\mathcal{R}^2$. This is therefore a case where the regressors are constrained to a manifold. The distance between two regressors in the regressor space can now be measured in two ways: the Euclidean $\mathcal{R}^2$ distance between points, and the geodesic distance measured along the curve, the manifold path. It is clear that the output, the altitude, is not a smooth function of regressors in the Euclidean space, since the altitudes vary substantially as the airplane comes back close to the earlier positions during climbing. However, if we use the geodesic distance in the regressor space, the altitude varies smoothly with regressor distance.

To see what the consequences are for predicting altitudes, suppose that for some reason, altitude measurements were lost for 8 consecutive time samples shortly after takeoff. To find a prediction for the missing measurements, the average of the three closest (in the regressor space, measured with Euclidean distance) altitude measurements were computed. The altitude prediction for one of the regressors is shown in Figure 2.5. The airplane turned and flew back on almost the same path as it just had flown, the three closest estimation regressors will

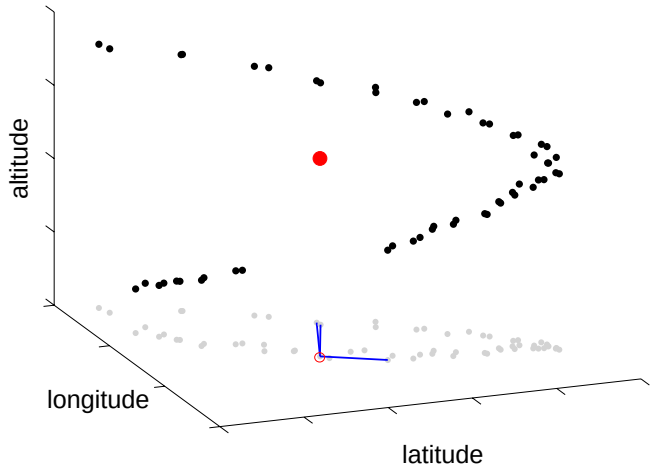

Figure 2.5: *The prediction of a missing altitude measurement (big filled circle). The encircled dot shows the position for which the prediction was computed. The three lines show the path to the three closest estimation regressors.*

therefore sometimes come from both before and after the turn. Since the altitude is considerably larger after the turn, the predictions will for some positions become heavily biased. In this case, it would have been better to use the three closest measurements along the flown path of the airplane. The example also motivates the semi-supervised smoothness assumption in regression.

Under the semi-supervised smoothness assumption, regression algorithms can be aided by incorporating the knowledge of a manifold. High-dimensional regression methods therefore have been modified to make use of the manifold and to estimate it (Belkin et al., 2006; Yang et al., 2006; Ohlsson et al., 2007). Since the regressors themselves contain information concerning the manifold, some regression methods use both regression-output pairs and regressors. This type of method is called *semi-supervised regression* or *semi-supervised modeling methods*. In contrast, in *supervised modeling* a relation between regressors and outputs is sought using a number of examples thereof *i.e.,* regression-output pairs. Most regression methods in system identification are supervised modeling methods. In *unsupervised modeling* the situation is rather different. Only one quantity is considered there and the task is rather to find patterns in the set of observations of this quantity. Semi-supervised modeling can be seen as a combination of supervised and unsupervised modeling.

2.11 Manifold Learning

Manifold learning is a fairly new research area aimed at finding, as the name suggests, descriptions of data on manifolds or intrinsic descriptions. The area has its roots in machine learning, and is a special form of *nonlinear dimensionality reduction* or *nonlinear feature extraction*. Some of the best known manifold learning algorithms are *isomap* (Tenenbaum et al., 2000), *Locally Linear Embedding* (LLE, Roweis and Saul (2000), discussed in the following section), *Laplacian eigenmaps* (Belkin and Niyogi, 2003) and *Hessian eigenmaps* (HLLE, Donoho and Grimes (2003)).

All manifold learning algorithms take as input a set of points sampled from some unknown manifold. The points are then expressed in a parameterization of the manifold, an intrinsic description (a set of points of the same dimension as the manifold), by searching for a set of new points preserving certain properties of the data. For example, Laplacian eigenmaps tries to preserve the Euclidean distance between neighboring points. Isomap tries to preserve the geodesic distances *i.e.,* the distance along the manifold, between points and locally linear embedding and Hessian eigenmaps make assumptions about local linearity and point neighborhoods which are aimed to be preserved. Manifold learning algorithms are unsupervised algorithms and most will not give an explicit expression for the map between high-dimensional points and their associated parameterization values.

2.11.1 Locally Linear Embedding

For finding intrinsic descriptions of data on a manifold, the manifold learning technique *Locally Linear Embedding* (LLE) can be used. LLE is a manifold learning technique which aims at preserving neighbors. In other words, given a set of points $\{\varphi_t\}_{t=1}^N$ residing on some n_z-dimensional manifold in $\mathcal{R}^{n_\varphi}$, LLE aims to find a new set of coordinates $\{z_1, \ldots, z_N\}$, $z_i \in \mathcal{R}^{n_z}$, satisfying the same neighbor-relations as the original points. The LLE algorithm can be divided into two steps:

Step 1: Define the w_{ij}s

Given data consisting of N real-valued vectors φ_i of dimension n_φ, the first step minimizes the cost function

$$\varepsilon(w) = \sum_{i=1}^{N} \left\| \varphi_i - \sum_{j=1}^{N} w_{ij}\varphi_j \right\|_2^2 \tag{2.58a}$$

with respect to w under the constraints

$$\begin{cases} \sum_{j=1}^{N} w_{ij} = 1, \\ w_{ij} = 0 \text{ if } \|\varphi_i - \varphi_j\|_2 > C_i(K) \text{ or if } i = j. \end{cases} \tag{2.58b}$$

Here, $C_i(K)$ is chosen so that only K weights w_{ij} become nonzero for every i. In the basic formulation of LLE, the number K and the choice of lower dimension $n_z \leq n_\varphi$ are the only design parameters, but it is also common to add a regularization

$$F_r(w) \triangleq \frac{r}{K} \sum_{i=1}^{N} [w_{i1}, \ldots, w_{iN}] \begin{bmatrix} w_{i1} \\ \vdots \\ w_{iN} \end{bmatrix} \sum_{j:w_{ij}\neq 0}^{N} \|\varphi_j - \varphi_i\|_2^2 \tag{2.59}$$

to (2.58a), see de Ridder and Duin (2002); Roweis and Saul (2000).

Step 2: Define the z_{ij}s

In the second step, w is now fixed. Let z_i be of dimension n_z and minimize

$$\Phi(z) = \sum_{i=1}^{N} \left\| {}_2 z_i - \sum_{j=1}^{N} w_{ij} z_j \right\|^2 \tag{2.60a}$$

with respect to $z = [z_1, \ldots, z_N]$, and subject to

$$\frac{1}{N} \sum_{i=1}^{N} z_i z_i^T = I \tag{2.60b}$$

using the weights w_{ij} computed in the first step. The solution z to this optimization problem is the desired set of n_z-dimensional coordinates which will work as an intrinsic description of the manifold. By expanding the squares we can rewrite

$\Phi(z)$ as

$$\Phi(z) = \sum_{i,j}^{N} (\delta_{ij} - w_{ij} - w_{ji} + \sum_{l}^{N} w_{li} w_{lj}) z_i^T z_j \tag{2.61a}$$

$$\triangleq \sum_{i,j}^{N} M_{ij} z_i^T z_j = \sum_{k}^{n_z} \sum_{i,j}^{N} M_{ij} z_{ki} z_{kj} = Tr(zMz^T) \tag{2.61b}$$

with M a symmetric $N \times N$ matrix with the ijth element

$$M_{ij} = \delta_{ij} - w_{ij} - w_{ji} + \sum_{l}^{N} w_{li} w_{lj}. \tag{2.62}$$

The solution to (2.60) is obtained by using *Rayleigh-Ritz theorem*, see *e.g.,* Horn and Johnson (1990, p. 176).

Theorem 2.2. *With Φ given by* (2.61)*, M by* (2.62) *and with v_i the unit length eigenvector of M associated with the ith smallest eigenvalue,*

$$\left[\begin{array}{c} v_1, \ldots, v_{n_z} \end{array} \right]^T = \arg\min_z \Phi(z) \quad s.t.\ zz^T = NI. \tag{2.63}$$

Remark 2.4. Notice that no explicit mapping is given, but more so an algorithm for computing an intrinsic description. If new points are introduced, the algorithm has to be rerun causing the intrinsic description for the old points to change.

The following example demonstrates how manifold learning or nonlinear feature extraction can be used in regression.

Example 2.7: Climate Reconstruction Cont'd

Let us now return to the climate reconstruction example in the introductory chapter, Example 1.1. Let us consider 10 shells grown in Belgium (see Ohlsson et al. (2009) for details). Since the temperature in the water had been monitored for these shells, this data set provides excellent means to test the ability to predict water temperature from chemical composition measurements. For these shells, the chemical composition measurements had been taken along the growth axis of the shells and paired up with temperature measurements. Between 30 and 52 chronologically ordered measurement were provided from each shell, corresponding to a time period of a couple of months.

Measurements from five of these shells are shown in Figure 2.6. The figure shows measurements of the relative concentrations of Sr/Ca, Mg/Ca and Ba/Ca (Pb/Ca is also measured, but not shown in the figure). The line shown between measurements connects the measurements coming from a shell and gives the chronological order of the measurements (two in time following measurements are connected by a line). As seen in the figure, measurements are highly restricted to a small region in the measurement space. Also, the water temperature (gray level coded in Figure 2.6) varies smoothly in the high-density regions. This together with that it is a biological process generating data, motivates the semi-

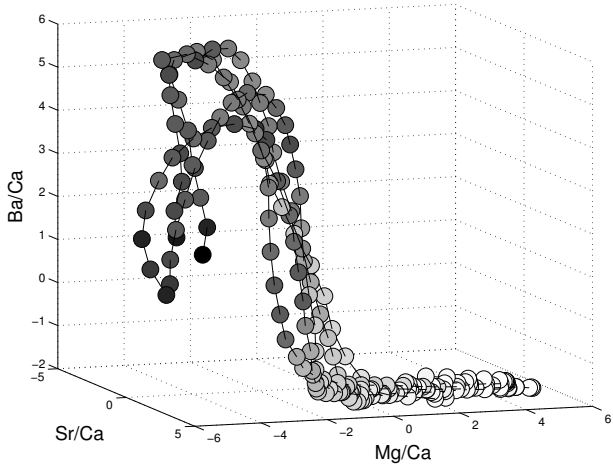

Figure 2.6: *A plot of the Sr/Ca, Mg/Ca and Ba/Ca concentration ratio measurements from five shells. Lines connects measurements (ordered chronologically) coming from the same shell. The temperatures associated with the measurements were color coded and are shown as different gray scales on the measurement points.*

supervised smoothness assumption when trying to estimate water temperature (outputs) from chemical composition measurements (4-dimensional regressors). Let us assume that the regressors are constrained to a one-dimensional manifold. LLE can then be applied to the regressors of the 10 shells to give a parameterization of the assumed one-dimensional manifold, an intrinsic description. This intrinsic description plotted against the measured water temperature is shown in Figure 2.7.

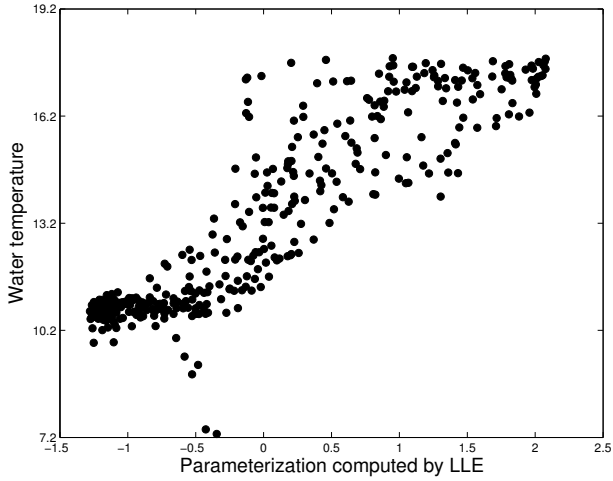

Figure 2.7: *The regressors (expressed using an intrinsic description) plotted against the measured water temperature. The intrinsic description was computed by using LLE.*

As seen in Figure 2.7, a linear estimate in the LLE parameterization would achieve a reasonably good estimate of the temperature.

2.12 Conclusion

This chapter served as an introduction to mathematical modeling and regression and introduced the fundamental knowledge and the necessary notation for the subsequent chapters. Several of the topics discussed are further discussed in papers of Part II. For example, impulse response identification discussed in Example 2.3 is the topic of Paper F and high dimensional regression, manifolds and manifold learning are discussed in Paper E.

3

State Estimation

Dynamic systems are characterized by that their output depends on current and past inputs. The effect that these inputs have had on the system is gathered in the *state*, which contains valuable information for *e.g.,* controllers and for decision making. It is a common situation that only parts of the state can be measured. Methods for recovering the full state of a dynamic system from these measurements are referred to as *state estimation techniques*. State estimation techniques use models to interpret the measured information.

3.1 The Standard Linear State-Space Model

The discrete-time standard linear state-space model with stochastic disturbances (see *e.g.,* Kailath et al. (2000, p. 161)) is given by

$$\begin{aligned} x_{t+1} &= A_t x_t + B_t u_t + G_t v_t, \\ y_t &= C_t x_t + e_t, \end{aligned} \tag{3.1a}$$

where x is the state, u a known input, v process noise, y the output and e the measurement noise. t index time. The process noise v and measurement noise e are here assumed to be zero mean *white noises* (see *e.g.,* Kailath et al. (2000, p. 4)): sequences of independent random vectors

$$\begin{aligned} \mathsf{E}[v_t] &= 0, & \mathsf{E}[e_t] &= 0 \quad \forall t \\ \mathsf{E}[v_t v_s^T] &= 0, & \mathsf{E}[e_t e_s^T] &= 0 \quad \text{if } t \neq s \\ \mathsf{E}[v_t v_t^T] &= Q_t, & \mathsf{E}[e_t e_t^T] &= R_t. \end{aligned} \tag{3.1b}$$

The independence of the noise sequences is required in order to make x_t a *Markov process*.

The model (3.1) with the process noise v being Gaussian is a standard model for control applications. v then represents the combined effect of all those non-measurable inputs that in addition to u affect the states. However, an equally common situation is that v corresponds to an *unknown input*. It could be

- a *load disturbance e.g.,* a step change in moment load of an electric motor, a (up or down) hill for a vehicle, *etc.* (Sometimes, the term load disturbance is used only for the case $B_t = G_t$.)
- an event that causes the state to jump, *a change*, see *e.g.,* Gustafsson (2001).

Such unknown inputs are not naturally modeled as Gaussian noise. Instead it is convenient to capture their unpredictable nature by (*cf.* eq (2.10)-(2.11) in Ljung (1999))

$$v_t = \delta_t \eta_t, \tag{3.2}$$

where (not to be confused with the Dirac delta function denoted by $\delta(\cdot)$)

$$\delta_t \triangleq \begin{cases} 0 & \text{with probability } 1-\mu, \\ 1 & \text{with probability } \mu, \end{cases} \qquad \eta_t \sim N(0, Q). \tag{3.3}$$

This makes $Q_t = \mu Q$ in (3.1b). The matrices A_t and G_t in (3.1a) may further model the waveform of the disturbance as a response to the pulse in v. Notice that if δ_t is known, v_t is Gaussian while an unknown δ_t leads to a non-Gaussian distributed v_t.

Example 3.1: DC Motor with Unknown Torque Load

Consider the discrete time model of a DC motor (see *e.g.,* Ljung (1999, pp. 95-97), $T_s = 0.1$ s, $\tau = 0.286$, $\beta = 40$)

$$\begin{aligned} x_{t+1} &= \begin{bmatrix} 0.7047 & 0 \\ 0.08437 & 1 \end{bmatrix} x_t + \begin{bmatrix} 11.81 \\ 0.6250 \end{bmatrix} (u_t + v_t), \\ y_t &= \begin{bmatrix} 0 & 1 \end{bmatrix} x_t + e_t. \end{aligned} \tag{3.4}$$

Here, x contains the angle and angular velocity of the motor shaft, y is noisy measurements of the motor shaft angle and u the applied voltage. The process noise v models a torque disturbance or an unknown torque load. Assuming that v is Gaussian is probably a bad assumption and in most applications a more sound assumption for v would probably be to model the process noise as in (3.2). The process noise v could also be set to pass through an integrator to model step changes.

We will get back to this example in Paper C and estimate the state x from the observed output y.

Example 3.2: Target Tracking

In *target tracking*, the goal is to estimate the state of a object given a number of sensor measurement. The object could be an airplane and the measurements, radar measurements, or it could be magnetometers placed in a crossing to track cars passing.

It is common to assume a dynamic motion model to model the kinematics of the object. The *continuous-time constant acceleration model* (see Chapter 13 in Gustafsson (2010)),

$$\begin{aligned} \dot{x}_t &= \begin{bmatrix} 0 & I_n & 0 \\ 0 & 0 & I_n \\ 0 & 0 & 0 \end{bmatrix} x_t + \begin{bmatrix} 0 \\ 0 \\ I_n \end{bmatrix} v_t, \\ y_t &= \begin{bmatrix} I_n & 0 & 0 \end{bmatrix} x_t + e_t, \end{aligned} \tag{3.5}$$

is a common choice. The state x contains the position, velocity and acceleration in n dimensions. The output y contains position measurements. The process noise v, the jerk (the derivative of the acceleration), is unknown and models the combined effect of all inputs that affect the state. e is the measurement noise of the sensor. The measurement noise e may very well be modeled by a Gaussian random variable. The lumped unknown inputs of the object gathered in v, however, is probably better modeled by *e.g.,* a piecewise constant signal. A piecewise constant signal is obtained by integrating a sequence of Dirac delta functions, this is illustrated in Figure 3.1.

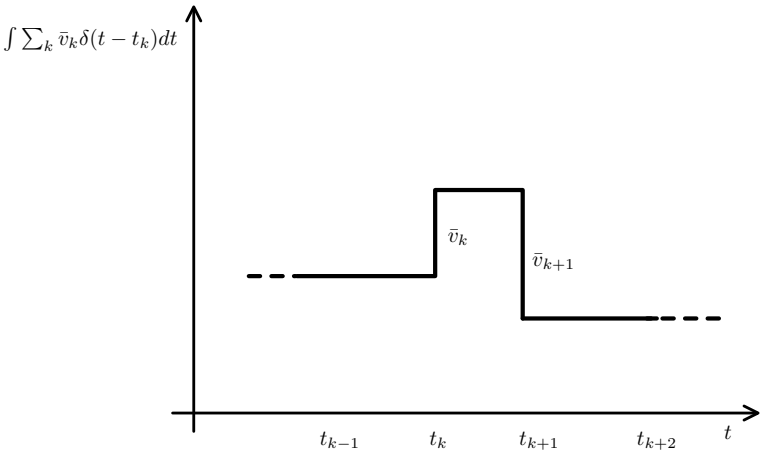

Figure 3.1: *Illustration of how a piecewise constant signal is obtained by integrating a sequence of Dirac delta functions. In this particular example there are impulses at t_k and t_{k+1} of sizes $\bar{v}_k$ and $\bar{v}_{k+1}$. These cause shifts of $\bar{v}_k$ and $\bar{v}_{k+1}$ at t_k and t_{k+1}.*

We can formulate this as

$$\begin{aligned}\dot{x}_t &= \begin{bmatrix}0 & I_n & 0 & 0\\ 0 & 0 & I_n & 0\\ 0 & 0 & 0 & I_n\\ 0 & 0 & 0 & 0\end{bmatrix} x_t + \begin{bmatrix}0\\0\\0\\I_n\end{bmatrix} \sum_k \bar{v}_k \delta(t - t_k),\\ y_t &= \begin{bmatrix}I_n & 0 & 0 & 0\end{bmatrix} x_t + e_t.\end{aligned} \tag{3.6}$$

Discretizing (3.6) with a sampling time $T_s = 0.1$ and under the assumption that $t_k = r_k T_s$, $r_k \in \mathcal{Z}$, give the discrete-time model (use *e.g.,* `sysd=c2d(sysc,Ts,'imp')` in MATLAB)

$$\begin{aligned}x_{kT_s+T_s} &= \begin{bmatrix}I_n & 0.1I_n & 0.005I_n & 0.0002I_n\\ 0 & I_n & 0.1I_n & 0.005I_2\\ 0 & 0 & I_n & 0.1I_n\\ 0 & 0 & 0 & I_n\end{bmatrix} x_{kT_s} + \begin{bmatrix}0.0002I_n\\0.005I_n\\0.1I_n\\I_n\end{bmatrix} \bar{v}_k,\\ y_{kT_s} &= \begin{bmatrix}I_n & 0 & 0 & 0\end{bmatrix} x_{kT_s}.\end{aligned} \tag{3.7}$$

To model $\bar{v}_k$ using the distribution given in (3.2) is now a good choice.

The relation between the ARX model (see (2.4)) and the state space model should be made clear. If we identify

$$x_t \leftrightarrow \theta_t,\ C_t \leftrightarrow \varphi_t^T,\ A_t \leftrightarrow I,\ B_t \leftrightarrow 0,\ G_t \leftrightarrow I, \tag{3.8}$$

the state space equation (3.1a) takes the form

$$\begin{aligned}\theta_{t+1} &= \theta_t + v_t,\\ y_t &= \varphi_t^T \theta_t + e_t,\end{aligned} \tag{3.9}$$

which is an ARX model with time varying parameters. This link between linear regression and state-space models is very well known, and described *e.g.,* in the classical survey by Åström and Eykhoff (1971). Possible knowledge of the parameter variations can be captured in more refined choices of A_t and G_t. θ is in (3.9) a *random walk*. If v is Gaussian, a (slowly) drifting model is described. For Gaussian noise v, the model (3.9) has been used to devise good tracking algorithms, *e.g.,* Section 11.6 in Ljung (1999). A piece-wise constant θ corresponds to a v as in (3.2) and that will be further discussed in Paper A.

3.2 State Estimation

Let us consider the estimation of x_t based on a set Y of the observations $\{y_t\}_{t=1}^N$. Write the estimate as

$$\hat{x}_t = F(Y). \tag{3.10}$$

There are two conceptually different cases:

- $\hat{x}_t$ is restricted to be a function of measurement up to and including time

t *i.e.,* $Y = \{y_t, y_{t-1}, y_{t-2}, \dots\}$. The estimation process is then referred to as *filtering*.

- $\hat{x}_t$ is based on measurements taken up to, including and later than time t *i.e.,* $Y = \{y_1, \dots, y_{t+1}, y_t, y_{t-1}, \dots, y_N\}$. The process of estimating x_t is then referred to as *smoothing*.

It is also common to distinguish between *linear* and *nonlinear filters* and *smoothers*. In linear filtering and smoothing F is a linear function of the elements in Y (and the initial state estimate). For a nonlinear filtering and smoothing algorithm, F is nonlinear in the elements of Y.

Two useful quantities when discussing filtering and smoothing are *bias* and *variance* of the estimate. A state estimate is said to be conditionally *unbiased* if

$$\mathsf{E}_{x_t}[\hat{x}_t - x_t|Y] = 0 \tag{3.11}$$

and otherwise conditionally *biased*. Note that this is equivalent to $\mathsf{E}_{x_t}[x_t|Y] = \hat{x}_t$. The conditional *covariance* of the estimate is given by

$$\mathsf{E}_{x_t}\left[\left(\hat{x}_t - x_t\right)\left(\hat{x}_t - x_t\right)^T\Big|Y\right]. \tag{3.12}$$

Alternatively, Y could be considered unknown and the expectations carried out over this quantity also. The state estimate is then said to be (unconditionally) unbiased if

$$\mathsf{E}_{x_t,Y}[\hat{x}_t - x_t] = 0. \tag{3.13}$$

The (unconditioned) covariance of the estimate is

$$\mathsf{E}_{x_t,Y}\left[\left(\hat{x}_t - x_t\right)\left(\hat{x}_t - x_t\right)^T\right]. \tag{3.14}$$

For the discrete-time standard linear state-space model with stochastic disturbances (3.1), the *Best Linear Unbiased Estimator* (BLUE) is given by the *Kalman Filter* (KF, Kalman (1960)) or *smoother* (*e.g.,* Kailath et al. (2000, p. 387)). We next give an introduction to the Kalman smoother and explain what "best" in "best linear unbiased estimator" refers to. We will only handle the smoothing case and not discuss filtering.

3.3 Kalman Smoother

In this thesis it is of interest to view the Kalman smoother as an explicit minimization problem. To arrive at the optimization formulation of the Kalman smoother, let first $\{y_t\}_{t=1}^N$ be a given set of observations satisfying (3.1) and let the initial state x_0 be a random variable independent of the noises e and v. Then, from (3.1b) it follows that the joint probability distribution can be written as

$$p\left(\{e_t\}_{t=1}^N, \{v_t\}_{t=1}^N, x_0\right) = p(x_0)\Pi_{t=1}^N p_e(e_t)p_v(v_t). \tag{3.15}$$

Assume now that $x_0 \sim N(0, \Gamma)$ and that e_t, v_t, $t = 1, \ldots, N$ are Gaussian distributed. Then (3.15) can be rewritten as

$$p\Big(\{e_t\}_{t=1}^N, \{v_t\}_{t=1}^N, x_0\Big) \propto e^{-\frac{1}{2}\|\Gamma^{-1/2}x_0\|_2^2} e^{-\frac{1}{2}\sum_{t=1}^N \|Q_t^{-1/2}e_t\|_2^2} e^{-\frac{1}{2}\sum_{t=1}^N \|R_t^{-1/2}v_t\|_2^2}. \tag{3.16}$$

Since, for $t = 1, \ldots, N$,

$$v_t = x_{t+1} - A_t x_t - B_t u_t, \quad e_t = y_t - C_t x_t, \tag{3.17}$$

(3.16) can be rewritten in terms of $\{x_t\}_{t=0}^N$ and $\{y_t\}_{t=1}^N$ as

$$\begin{aligned}\log p\Big(\{y_t\}_{t=1}^N \Big| \{x_t\}_{t=0}^N\Big) \propto &-\Big\|\Gamma^{-1/2}x_0\Big\|_2^2 - \sum_{t=1}^N \Big\|R_t^{-1/2}(y_t - C_t x_t)\Big\|_2^2 \\ &-\Big\|Q_{t-1}^{-1/2}(x_t - Ax_{t-1} - Bu_{t-1})\Big\|_2^2.\end{aligned} \tag{3.18}$$

Maximizing this quantity with respect to $\{x_t\}_{t=0}^N$ leads to the maximum likelihood estimate (MLE) for $\{x_t\}_{t=0}^N$. The MLE for $\{x_t\}_{t=0}^N$ can equivalently be written as

$$\underset{x_t, t=0,\ldots,N}{\arg\min} \|\Gamma^{-1/2}x_0\|_2^2 + \sum_{t=1}^N \Big\|R_t^{-1/2}(y_t - C_t x_t)\Big\|_2^2 + \Big\|Q_{t-1}^{-1/2}(x_t - A_{t-1}x_{t-1} - B_{t-1}u_{t-1})\Big\|_2^2 \tag{3.19}$$

which is recognized as the classical Kalman smoothing estimate, *e.g.,* Kailath et al. (2000, p. 387). Note that (3.19) is a (ℓ_2-regularized) least squares problem. The solution can therefore be shown to be linear in $\{y\}_{t=1}^N$ (and x_0). The solution is usually given in various recursive filter forms, see *e.g.,* Ljung and Kailath (1976).

When all densities are Gaussian (e_t, v_t, x_0 Gaussian), (3.19) gives the best unbiased estimate (among both linear and nonlinear estimators) since no other unbiased estimator can obtain a smaller variance. That is, let $\hat{x}_t$ be the Kalman estimate and let $\bar{x}_t$ be any other unbiased state estimate. Then, with expectation over both x_t and Y,

$$\mathsf{E}_{x_t,Y}\Big[(\bar{x}_t - x_t)(\bar{x}_t - x_t)^T\Big] - \mathsf{E}_{x_t,Y}\Big[(\hat{x}_t - x_t)(\hat{x}_t - x_t)^T\Big] \geq 0. \tag{3.20}$$

This also implies that no other unbiased estimator can obtain a lower MSE *i.e.,*

$$tr\,\mathsf{E}_{x_t,Y}\Big[(\hat{x}_t - x_t)(\hat{x}_t - x_t)^T\Big] = \mathsf{E}_{x_t,Y}\Big[(\hat{x}_t - x_t)^T(\hat{x}_t - x_t)\Big]. \tag{3.21}$$

(3.20) and (3.21) also hold if the expectations is taken w.r.t x_t and conditional on Y. It further holds that x_t given $\{y_t\}_{t=1}^N$ is Gaussian (the mean given by $\hat{x}_t$, *i.e.,* $\hat{x}_t = \mathsf{E}[x_t|y_1, \ldots, y_N]$).

If e_t, v_t or x_0 is not Gaussian, the Kalman smoother is still the best unbiased linear estimator. That means that we can not do better than using a Kalman smoother if v is distributed as (3.2), the sequence δ_t is unknown and the smoother is restricted to be linear. If we knew the δ_t-sequence (and the measurement noise was Gaussian), the Kalman smoother would be the best estimator among both linear and nonlinear estimators, since all noises would be Gaussian (with time varying noise covariance).

See Anderson and Moore (1979, Chap. 7) for more on the Kalman smoother and its properties.

3.4 Kalman Filter (Smoother) Banks

Based on the process noise model (3.2), a number of nonlinear methods have been developed. If δ_t is unknown, we could hypothesize in each time step that it is either 0 or 1. This leads to a large bank (2^N) of Kalman smoothers as the optimal solution. The posterior probability of each smoother can be estimated from this bank, which consists of a weighted sum of the state estimates from each smoother. See Chapter 10 in Gustafsson (2010) for more on smoother banks.

In practice, the number of smoothers in the bank must be limited due to computational limitations, and there are two main options (see Chapter 10 in Gustafsson (2010)):

- Merging trajectories of different δ_t sequences. This includes the well known *Interacting Multiple Model* filter (IMM filter, Blom and Bar-Shalom (1988)).
- Pruning, where unlikely sequences are deleted from the filter bank.

3.5 Conclusion

This chapter gave a brief introduction to filtering and smoothing. We continue the discussion on smoothing and impulsive process noise in Paper C. In particular we explore the fact that the sequence generated by (3.2), arranged as a vector, contains elements identical to zero, it will be a *sparse* vector. This leads us to the concept of sparseness and regularization for sparseness. Sparseness and regularization for sparseness are discussed in the next chapter, Chapter 4.

4

Regularization for Sparseness

Sparseness is all about zeros. A matrix or vector is said to be *sparse* if it contains a relatively large number of zeros. If a quantity is given to be sparse, it is often a computational remedy *e.g.,* when solving equation systems or in optimization. However, sparsity has also shown great importance for other reasons, in *e.g.,* statistical learning and signal processing. The hype around sparsity in statistical learning is mostly due to the success of *lasso* (least absolute shrinkage and selection operator, Tibsharani (1996); Chen et al. (1998), see also Hastie et al. (2001, p. 64)) and in signal processing sparsity has got attention due to the sampling protocol *Compressed Sensing* (CS, Donoho (2006); Candès et al. (2006)).

Formally, sparse is defined as (see *e.g.,* Zibulevsky and Elad (2010)):

Definition 4.1 (Sparse). A vector $z \in \mathcal{R}^n$ is said to be sparse if

$$\|z\|_0 \ll n. \tag{4.1}$$

$\|\cdot\|_0$ here denotes the zero (quasi-)norm. The zero norm is the number of nonzero elements of a vector (see Appendix A).

4.1 When is Sparsity a Desirable Property?

Sparsity is wanted in various situations. Sparsity can *e.g.,* be used for variable selection, as in lasso, for image denoising and filter design as in Starck et al. (2002); Bioucas-Dias (2006) or as a sample protocol, as in compressed sensing. What the above applications have in common is that the underlying problem has a combinatorial nature. The problem could *e.g.,* be to select a subset of variables, basis

functions, times instances *etc.* that solves some problem in an optimal manner.

The following three examples give a flavor for when, where and how sparseness can be used. We will revisit these examples at later points of the chapter as well.

Example 4.1: Lasso

Consider the task of estimating a linear regression model

$$f(\varphi, \theta) = \varphi^T \theta. \tag{4.2}$$

Assume that an estimation data set $\{(\varphi_t, y_t)\}_{t=1}^{N_e}$, $y_t \in \mathcal{R}$, $\varphi_t \in \mathcal{R}^{n_\varphi}$ is given for this purpose. Also assume that $n_\varphi > N_e$. Minimizing the sum of squared residuals

$$\sum_{t=1}^{N_e} (y_t - \varphi_t^T \theta)^2 \tag{4.3}$$

to determine θ leads to an ill-posed problem (see Example 2.2). In particular, the solution will not be unique. We saw previously how ℓ_2-regularization (see Example 2.2) can be used to transform (4.3) into a well-posed problem

$$\min_\theta \sum_{t=1}^{N_e} (y_t - \varphi_t^T \theta)^2 + \lambda \|\theta\|_2^2, \quad \lambda \in \mathcal{R}^+. \tag{4.4}$$

The ℓ_2-regularization added in (4.4) favors small $\|\theta\|_2^2$. However, typically all θ-elements turn out non-zero and it may therefore be difficult to understand which regressor elements that are meaningful. Besides, one also needs to continue to acquire the whole regressor vector φ to use the model. If each element in φ_t requires a measurement to be done, acquiring the whole regressor vector may be impractical if n_φ is large.

The idea of lasso is to find a regression parameter θ so that the model (4.2) gives a good fit to the estimation data *i.e.,* makes

$$\sum_{t=1}^{N_e} (y_t - \varphi_t^T \theta)^2 \tag{4.5}$$

small and at the same time obtain a θ which is sparse. The sparsity constraint will cause a large number of θ-elements to be zero. Lasso therefore gives the possibility to interpret and say what regression elements that are meaningful for a good prediction result. Zeros in θ mean that the associated regressor elements are not needed, time and money can therefore be saved by only measuring the φ-elements associated with non-zero θ-elements. The idea of lasso leads to a criterion

$$\min_\theta \sum_{t=1}^{N_e} (y_t - \varphi_t^T \theta)^2 + \lambda \|\theta\|_0, \quad \lambda \in \mathcal{R}^+. \tag{4.6}$$

We will come back to lasso and the mathematical details in Example 4.4.

Example 4.2: Compressed Sensing Cont'd

Let us return to the discussion of audio compression and sampling given in the introductory part of the thesis, Example 1.3. We there argued that it was rather meaningless to measure a lot of information if 90% will be thrown away before someone even listened to the song, or as Donoho (2006) wrote,

> "Why go to so much effort to acquire all the data when most of what we get will be thrown away? Can we not just directly measure the part that will not end up being thrown away?"

In an MP3 encoder, the audio stream is divided into several frequency bands. The audio of a frequency band is then discarded if it is weaker than some certain threshold (Brandenburg, 1999; Hayes, 2009). The problem is that even though an audio recording can well be represented using the audio in a small number of frequency bands, we do not know what bands that are going to be discarded before we start sample. We therefore need to sample all frequency bands and then compress and throw away a major part of our sampled data. This was what many thought before compressed sensing was introduced in 2006.

Let $x \in \mathcal{R}^{n_x}$ be a quantity that we are interested in. In *compressed sensing* (also known as *compressive sensing*, *compressive sampling*, *compressed sampling*) it is assumed that the signal x is composed of a very limited number of *atoms* from a *dictionary* containing a large number of typical signal shapes or *basis functions*. Let these signal shapes be columns in the matrix $A \in \mathcal{R}^{n_x \times n_z}$, typically $n_z \gg n_x$. The signal x is hence assumed to have the property

$$x = Az, \quad z \in \mathcal{R}^{n_z} \text{ sparse.} \tag{4.7}$$

A dictionary, or A, that has these properties is in compressed sensing assumed known. It could *e.g.*, be suitable to chose a dictionary containing sampled sine and cosine signals of difference frequencies if x contains a sequence of audio samples.

Remark 4.1. All signals that people find meaningful can be decomposed as in (4.7) (Hayes, 2009). A sequence of independent random numbers is an example of a signal that can not be decomposed using a sparse z.

Let $M \in \mathcal{R}^{n_y \times n_x}$, $n_x \gg n_y$ and define $y \in \mathcal{R}^{n_y}$ by

$$y \triangleq Mx = MAz. \tag{4.8}$$

What is important is that y has considerably lower dimmension than x. Hence, y can be seen as a compressed version of x. The idea of compressed sensing is now to measure y rather than x. That is, to measure a few linear combinations of the elements in x rather than x directly. This means that we should construct a number (n_y) of microphones that each give a sample which *e.g.*, is a weighted average of sounds during the last second. The microphones should not be identical, they all need to form different weighted averages. Assume also that the weights used to form these averages are known, that is, we know M.

We now have y, which we have acquired using less sampling and can store using

less memory space than x would have needed. How do we recover x so to be able to listen to the second of audio?

Since M, A and y are all known and z was assumed sparse, it is natural to seek for an estimate $\hat{z}$ using

$$\hat{z} = \arg\min_z \|z\|_0 \quad \text{s.t. } y = MAz. \tag{4.9}$$

We can then obtain $\hat{x}$ as $\hat{x} = A\hat{z}$. What is remarkable is that under certain rather mild assumptions on the matrices A and M and if z satisfies (4.7) and is sufficiently sparse, $\hat{x} = x$ (see *e.g.,* Bruckstein et al. (2009)). The audio can hence be perfectly recovered even though $n_x \gg n_y$!

We will return to compressed sensing in Example 4.5 and there present the mathematical details.

Remark 4.2 (Nyquist-Shannon Sampling Criterion and Compressed Sensing). The *Nyquist-Shannon sampling criterion* states that for a *bandlimited signal* (no energy above some certain frequency) the sampling frequency should be twice that of the bandlimit to guarantee the possibility to perfectly reconstruct the time-continuous signal (see *e.g.,* Oppenheim et al. (1996, p. 519)). With no further information, to use a sample frequency twice that of the bandlimit is actually the best thing to do (Tropp et al., 2010). However, if the signal is known to be *e.g.,* a combination of a few basis functions, a perfect reconstruction can be obtained at a lot lower sampling frequencies.

Example 4.3: The Huber Loss Function

Consider the following setup

$$y_t = \varphi_t^T \theta_0 + e_t + \tau_t, \quad y_t \in \mathcal{R},\ e_t \sim N(0, \sigma^2), \tag{4.10}$$

where $\theta_0 \in \mathcal{R}^{n_\theta}$ is an unknown vector and e_t the measurement noise. The scalar variable τ_t models an *outlier* and will therefore be zero for most t but occasionally non zero. Let $\{(\varphi_t, y_t)\}_{t=1}^{N_e}$ be a given estimation data set.

Desiring an estimate of θ_0, we can use the least squares estimate,

$$\hat{\theta}_{ls} = \arg\min_\theta \sum_{t=1}^{N_e} (y_t - \varphi_t^T \theta)^2 \tag{4.11a}$$

$$= \arg\min_\theta \sum_{t=1}^{N_e} (\varphi_t^T \theta_0 + e_t + \tau_t - \varphi_t^T \theta)^2. \tag{4.11b}$$

If $\tau_t \neq 0$ for some $t = 1, \ldots, N_e$, it is likely that the estimate of θ_0 is adjusted to fit these fluctuations in τ_t. We can try to get around this by estimating τ_t and then subtract the estimate from our measurements y_t.

As outliers, by definition, appear seldom, a realization of the time series $\{\tau_t\}_{t=1}^{N_e}$

arranged as a vector, will be a sparse vector. We are therefore led to consider

$$\min_{\theta,\eta_1,\eta_2,\ldots,\eta_{N_e}} \sum_{t=1}^{N_e} (y_t - \varphi_t^T\theta - \eta_t)^2 + \lambda \left\| [\eta_1\ \eta_2 \ldots \eta_{N_e}]^T \right\|_0, \tag{4.12}$$

for some $\lambda \in \mathcal{R}^+$. Here, $\{\eta_t\}_{t=1}^{N_e}$ serves as an estimate of the realization of $\{\tau_t\}_{t=1}^{N_e}$ associated with estimation data $\{(\varphi_t, y_t)\}_{t=1}^{N_e}$. We return to this example in Example 4.6.

Bruckstein et al. (2009); Zibulevsky and Elad (2010) further exemplify and motivate sparsity in signal processing and modeling.

4.2 Methods for Obtaining Sparsity

The ℓ_0-norm causes optimization problems to be non-convex and combinatorial. Solving the optimization problem (4.9)

$$\min_z \|z\|_0, \quad \text{s.t. } y = MAz, \tag{4.13}$$

e.g., boils down to an exhaustive combinatorial search: Fix all element in z except the first to zero and check if there is a z satisfying $y = MAz$. If not, continue by fixing all except the second element to zero and check if there is a z satisfying $y = MAz$. Go through the whole vector z if necessary, letting one element free and fixing all other to zero, one by one. If no z satisfying $y = MAz$ is found, go through different combinations of two nonzero elements in a search for a z satisfying $y = MAz$. And so on. See *e.g.,* Bruckstein et al. (2009). Not very surprising, (4.13) can actually be shown to be NP-hard (Natarajan, 1995).

The optimization problems (4.6) and (4.12) are of the form

$$\min_\theta \|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_0, \quad \lambda \in \mathcal{R}^+, \tag{4.14}$$

and are also in general NP-hard (Huo and Ni, 2007). If the measurements (4.8) in compressed sensing are noisy, an optimization problem of the form (4.14) replaces (4.9), see Candès et al. (2006). Note that many model selection criteria *e.g.,* AIC (see (2.10) for AIC) has also the form (4.14) for a linear regression model, see *e.g.,* Huo and Ni (2007).

The combinatorial optimization problem that (4.13) and (4.14) lead to is often impractical to solve and several approximation techniques have therefore been proposed.

Greedy algorithms (see *e.g.,* Tropp (2004); Bruckstein et al. (2009)) start with a z identical to zero (or θ identical to zero if (4.14) is considered). Greedy algorithms then let the element in z which *e.g.,* increases the fit the most free and estimate z. The greedy algorithm then continues by, one by one, letting the z element that leads to the best fit free and re-estimating z. The algorithm terminates when a good enough fit has been obtained. Under the assumption that the z solving

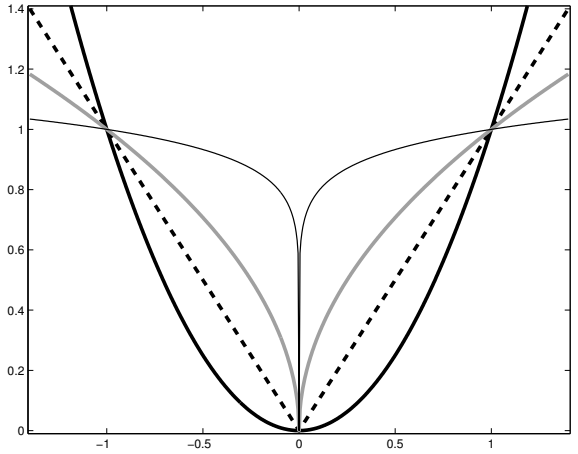

Figure 4.1: *For a one-dimensional variable, the (squared) ℓ_2-norm, $(\cdot)^2$, with solid black thick line, the ℓ_1-norm, $|\cdot|$, showed with dashed black line, $|\cdot|^{1/2}$ with gray line and $|\cdot|^{1/10}$ with solid thin black line.*

(4.13) is sufficiently sparse and MA sufficiently *incoherent* (see *e.g.,* Candès et al. (2010)) *i.e.,*

$$\max_{j<k} \frac{|(MA)(:,j)^T(MA)(:,k)|}{\|(MA)(:,j)\|_2\|(MA)(:,k)\|_2} \ll 1, \tag{4.15}$$

some greedy algorithms give the same solution as that of (4.13), see *e.g.,* Bruckstein et al. (2009). For the problem (4.14) the correct support can be guaranteed if the solution of (4.14) is sufficiently sparse, the signal to noise ratio is sufficiently good and Φ sufficiently incoherent, see *e.g.,* Bruckstein et al. (2009). Many variants of greedy algorithms exist. *Forward stepwise regression* (see *e.g.,* Daniel and Wood (1980, pp. 84-85), known as *matching pursuit* in signal processing, see *e.g.,* Mallat and Zhang (1993)) may be the one most known to the system identification community. However, also *e.g., Least Angle Regression* (LARS, Efron et al. (2004)) is a variant of greedy algorithm.

The *FOCUSS* (FOCal Underdetermined System Solver, see *e.g.,* Bruckstein et al. (2009)) method is another approximation method. In FOCUSS an approximation to (4.13) is sought by searching for a local minimum of the ℓ_p, $0 < p < 1$, regularized problem

$$\min_z \|z\|_p, \quad \text{s.t. } y = MAz. \tag{4.16}$$

This is an non-convex problems.

The "closest" convex problem to (4.13) and (4.14) is obtained by replacing the ℓ_0-norm with the ℓ_1-norm, see Figure 4.1. This is a *convex relaxation* of the problem. If (4.13) is relaxed by replacing the zero-norm with the ℓ_1-norm,

$$\min_z \|z\|_1, \quad \text{s.t. } y = MAz, \tag{4.17}$$

we obtain what is referred to as the *basis pursuit* (Chen et al., 1998). This problem can be solved using linear programming (see *e.g.,* Donoho (2006)).

If (4.14) is relaxed by replacing the zero-norm with the ℓ_1-norm,

$$\min_{\theta} \|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_1, \quad \lambda \in \mathcal{R}^+, \tag{4.18}$$

we obtain what is referred to as the *basis pursuit denoise* (Chen et al. (1998)) in the signal processing community and lasso in the statistical community. The problem given in (4.18) is a ℓ_1-regularized least squares problems. The next section discusses the usage of ℓ_1 regularization for obtaining sparsity.

4.3 ℓ_1-Regularization

ℓ_1-regularization is by no means a new concept (see Appendix I of Tropp (2006) for a historical review). In fact, it has been a regularization technique and a known way to obtain sparsity since the 1970s. It has gained a lot of popularity and publicity lately though.

A ℓ_1-regularized problem has the form

$$\min_{\theta} V(\theta) + \lambda\|\theta\|_1, \quad \lambda \in \mathcal{R}^+, \tag{4.19}$$

where V is the criterion of fit, $\|\cdot\|_1$ the ℓ_1-norm and λ is the regularization parameter. The criterion of fit $V(\theta)$ is often the least squares criterion $\|y - \Phi\theta\|_2^2$, as in (4.18), but there are many other interesting choices, *e.g.,* Riezler and Vasserman (2004); Chen et al. (2009).

For the ℓ_1-regularized least squares procedure ($V(\theta) = \|y - \Phi\theta\|_2^2$ in (4.19))

$$\min_{\theta} \|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_1, \tag{4.20}$$

it has been shown that the solution (for a proper value for λ) has the same zero elements (but possibly more) as the solution of

$$\min_{\theta} \|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_0, \tag{4.21}$$

if Φ is sufficiently incoherent (see (4.15)) and the measurement noise weakly correlated with Φ (Tropp, 2006). The solution may however not be unique, since (4.20) is not necessarily strictly convex, see *e.g.,* Bertsekas et al. (2003, Prop. 2.1.2)).

Example 4.4: Lasso Cont'd

Let us return to Example 4.1 and lasso. The idea in lasso is to find a θ so that the a linear model (4.2) gives a good fit to the estimation outputs and at the same time obtains a θ which is sparse. We formulated this as

$$\min_{\theta} \sum_{t=1}^{N_e} (y_t - \varphi_t^T\theta)^2 + \lambda\|\theta\|_0, \tag{4.22}$$

for some $\lambda \in \mathcal{R}^+$. This problem is combinatorial and a convex relaxation is therefore used to obtain the ℓ_1-regularized least squares, or the lasso, criterion

$$\min_{\theta} \sum_{t=1}^{N_e} (y_t - \varphi_t^T \theta)^2 + \lambda \|\theta\|_1. \tag{4.23}$$

The ℓ_1-regularization in lasso penalizes elements of θ different than zero and therefore causes elements of θ that do not provide a significant decrease of the fit term to be zero. The property of the ℓ_1-regularization that causes elements to become identical to zero, and not only small as in the ℓ_2-regularization, is discussed in Section 4.3.1.

The θ resulting from solving (4.23), let us say $\hat{\theta}$, will be biased since terms that do provide a better fit also are being penalized and dragged towards zero. The bias is often adjusted for by re-estimating the regression parameters according to

$$\min_{\theta} \sum_{t=1}^{N_e} (y_t - \varphi_t^T \theta)^2 \quad \text{s.t. } \theta(i) = 0 \text{ if } \hat{\theta}(i) = 0,\ i = 1, \ldots, dim(\hat{\theta}), \tag{4.24}$$

with $\hat{\theta}$ from (4.23). This makes the regression parameter unbiased if lasso correctly identified the zero elements in θ.

Example 4.5: Compressed Sensing Cont'd

We now return to Example 4.2 to carry out the mathematical details. We argued that to reconstruct x it was natural to seek for an estimate $\hat{z}$ using

$$\hat{z} = \arg\min_{z} \|z\|_0 \quad \text{s.t. } y = MAz, \tag{4.25}$$

and then obtain $\hat{x}$ as $\hat{x} = A\hat{z}$. Due to the combinatorial complexity of (4.25) we are led to consider *e.g.,* a convex relaxation, such as replacing the ℓ_0-norm with the ℓ_1-norm. If the measurements y are noisy, the equality constraint in (4.25) is removed and $\|y - MAz\|_2^2$ added to the objective function. We are led to consider the ℓ_1-regularized least-squares problem

$$\hat{z} = \arg\min_{z} \|y - MAz\|_2^2 + \lambda \|z\|_1, \quad \lambda \in \mathcal{R}^+. \tag{4.26}$$

What is remarkable is that with considerably fewer samples than what the Nyquist-Shannon sampling criterion would have told you to use and with the relaxed ℓ_0-norm, a close to perfect reconstruction of x can be obtained, see *e.g.,* Candès and Wakin (2008). In fact, it has been shown that compressed sensing is nearly as effective as if having an oracle telling us what elements of z that are nonzero and we would have measured only those (Candès and Wakin, 2008).

Example 4.6: The Huber Loss Function Cont'd

Let use finally return to Example 4.3. We there assumed that we got measurements $\{(\varphi_t, y_t)\}_{t=1}^{N_e}$ from

$$y_t = \varphi_t^T \theta_0 + v_t + \tau_t, \quad v_t \sim N(0, \sigma^2), \tag{4.27}$$

where τ_t modeled outliers and was assumed to be an in time sparse variable. The model parameter θ_0 is an unknown vector. Desiring an estimate of θ_0, we can use the least squares estimate,

$$\theta_{\text{ls}} = \arg\min_{\theta} \sum_{t=1}^{N_e} (y_t - \varphi_t^T \theta)^2 \tag{4.28a}$$

$$= \arg\min_{\theta} \sum_{t=1}^{N_e} (\varphi_t^T \theta_0 + e_t + \tau_t - \varphi_t^T \theta)^2. \tag{4.28b}$$

If $\tau_t \neq 0$ for some $t = 1, \ldots, N_e$, it is likely that the estimate of θ_0 is adjusted to fit these fluctuations in τ_t. We can try to get around this by estimating τ_t and then subtract the estimate from our measurements y_t. As we have assumed that τ_t is sparse, it is motivated to minimize

$$\sum_{t=1}^{N_e} (y_t - \varphi_t^T \theta - \eta_t)^2 + \lambda \left\| \begin{bmatrix} \eta_1 & \eta_2 & \ldots & \eta_{Ne} \end{bmatrix} \right\|_0, \quad \lambda \in \mathcal{R}^+, \tag{4.29}$$

with respect to the outlier estimate η and θ. Here, λ is seen as a design parameter that controls the sparsity of η. Using a convex relaxation, we arrive at the less computationally intensive ℓ_1-regularized least squares problem

$$\sum_{t=1}^{N_e} (y_t - \varphi_t^T \theta - \eta_t)^2 + \lambda \left\| \begin{bmatrix} \eta_1 & \eta_2 & \ldots & \eta_{Ne} \end{bmatrix} \right\|_1. \tag{4.30}$$

As shown in Appendix B, (4.30) is equivalent to

$$\sum_{t=1}^{N_e} \psi\big(y_t - \varphi_t^T \theta\big) \tag{4.31}$$

with

$$\psi(x) \triangleq \begin{cases} |x|^2, & \text{if } |x| < \lambda/2, \\ \lambda|x| - \lambda^2/4 & \text{otherwise.} \end{cases} \tag{4.32}$$

The function $\psi(\cdot)$ is called the *Huber loss function* or the *Huber norm* (Huber, 1973). The Huber loss function has been applied frequently within regression and classification since its introduction in the 1970s by Huber. Its popularity is due to its ability to reduce the affect of an outlier and thereby gain robustness to the algorithm. The Huber loss function, $\psi(\cdot)$, shown in Figure 4.2, is a hybrid between the ℓ_1 and the ℓ_2-norm. That the assumption of a sparse outlier τ here leads to the Huber loss function is rather intuitive but still illustrative.

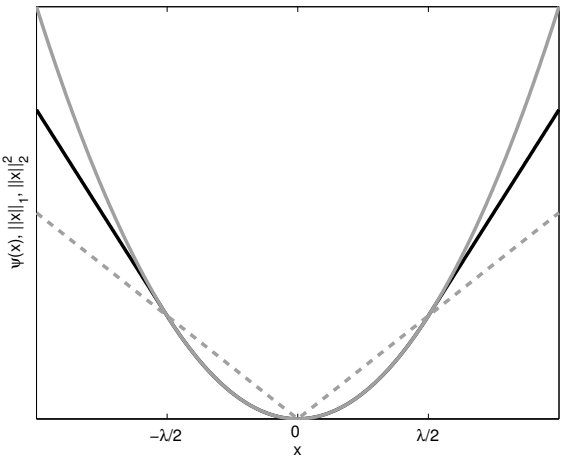

Figure 4.2: *The Huber loss function $\psi(x)$ plotted with thick solid black line for a one-dimensional x. The ℓ_1 and squared ℓ_2-norm are also shown, dashed and solid gray line, respectively.*

It is interesting to notice that minimizing

$$\sum_{t=1}^{N_e} (y_t - \varphi_t^T \theta)^2 + \lambda \|\theta\|_1, \tag{4.33}$$

$\lambda \in \mathcal{R}^+$, can be interpreted as a MAP estimate of a posterior distribution proportional to

$$e^{-\sum_{t=1}^{N_e} (y_t - \varphi_t^T \theta)^2 / 2\sigma^2} e^{-\lambda \|\theta\|_1 / 2\sigma^2}. \tag{4.34}$$

Using Bayes' theorem, see Theorem 2.1 on page 26, the first term in (4.34) can be interpreted as the likelihood

$$p\left(\{y_t\}_{t=1}^{N_e} \middle| \theta, \{\varphi_t\}_{t=1}^{N_e}\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\sum_{t=1}^{N_e} (y_t - \varphi_t^T \theta)^2 / 2\sigma^2} \tag{4.35}$$

associated with

$$y_t = \varphi_t^T \theta_0 + e_t, \quad e_t \sim N(0, \sigma^2). \tag{4.36}$$

The second term in (4.34) can be interpreted as a prior $p(\theta) = \frac{1}{4\sigma^2} e^{-\lambda \|\theta\|_1 / 2\sigma^2}$. The prior associated with the ℓ_1-regularization is hence $p(\theta) = \frac{1}{4\sigma^2} e^{-\lambda \|\theta\|_1 / 2\sigma^2}$. In the literature this is referred to as a *Laplace* or an *independent double exponential* prior (see *e.g.,* Hastie et al. (2001, p. 72)).

4.3.1 What Property of the ℓ_1-Regularization Causes Sparseness?

Let us investigate why ℓ_1-regularization causes sparseness. Consider

$$\min_{\theta} \|y - \Phi\theta\|_2^2 \quad \text{s.t. } \|\theta\|_1 \leq \eta. \tag{4.37}$$

This problem is identical to that of (4.20) in the sense that, for any $\lambda \in \mathcal{R}^+$, there exists a η ($= \|\theta^*\|_1$, where θ^* minimizes (4.20)) so that the minimizing θ is the same for (4.20) and (4.37). The *Karush-Kuhn-Tucker* (KKT, see *e.g.,* Boyd and Vandenberghe (2004, p. 244)) conditions can be used to show this.

Consider now the left of Figure 4.3. The gray square at the origin shows the

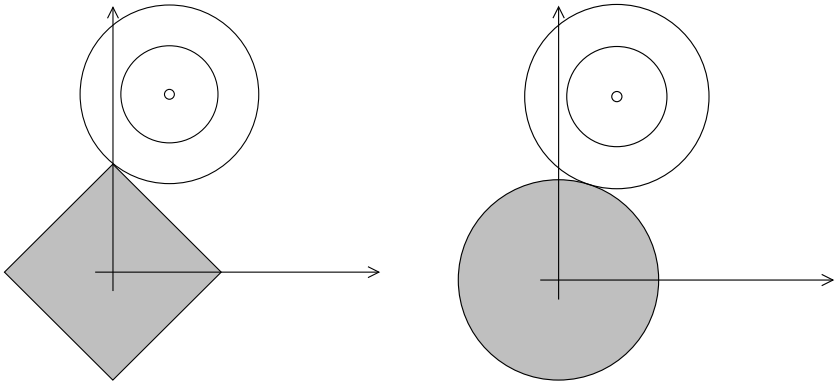

Figure 4.3: *Left figure: An illustration of $\|\theta\|_1 \leq \eta$ (gray area) and the level-curves of $\|y - \Phi\theta\|_2^2$. Right figure: An illustration of $\|\theta\|_2^2 \leq \eta$ (gray area) and the level-curves of $\|y - \Phi\theta\|_2^2$. In both the right and the left figure, $\|y - \Phi\theta\|_2^2$ is assumed to have a unique minimum. If $\|y - \Phi\theta\|_2^2$ does not have a unique minimum, there will be a continuum of points, on a line, minimizing $\|y - \Phi\theta\|_2^2$ and the level curves would be parallel to that line.*

neighborhood $\|\theta\|_1 \leq \eta$ for a two dimensional regressor (*i.e.,* $dim(\theta) = 2$). The level-curves of $\|y - \Phi\theta\|_2^2$ are also shown. These are depicted as circles (generally these level curves are ellipses) centered at $\arg\min_\theta \|y - \Phi\theta\|_2^2$. From the illustration it is seen that the θ minimizing (4.37) must be the θ-value at the intersection between the square and one of the level-curves. Note now that when this intersection happens on one of the axis, the optimal θ get one zero element. Try to move around the level-curves of $\|y - \Phi\theta\|_2^2$. Most choices gives an intersection at an axis. For a higher dimensional case ($dim(\theta)$ large), the gray square turns into a hyper-cube. When intersection happens on *e.g.,* one of the vertexes, the optimal θ has elements equal to zero and therefore turns out as sparse.

Consider now the right part of Figure 4.3. The right part illustrates what happens if the regularization is chosen as $\|\cdot\|_2^2$ (ridge regression, see Example 2.2) instead

of $\|\cdot\|_1$ as in the ℓ_1-regularization. Consider

$$\min_{\theta} \|y - \Phi\theta\|_2^2 \quad \text{s.t. } \|\theta\|_2^2 \leq \eta, \tag{4.38}$$

which for a particular choice of η gives the same solution as

$$\min_{\theta} \|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_2^2. \tag{4.39}$$

The gray circle now illustrates $\|\theta\|_2^2 \leq \eta$ which is a disc centered at the origin. The level-curves of $\|y - \Phi\theta\|_2^2$ are also shown, just as in the left of Figure 4.3. The solution to (4.38) can now be seen given by the intersection between the disc and a level-curve. Try to move the level-curves around, the intersection is this time very seldom on an axis. The minimizing θ will therefore generally not be sparse.

An Explicit Solution

For illustration, let us consider a special case which has an explicit solution. Consider

$$\min_{\theta} \|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_1, \tag{4.40}$$

where $\lambda \in \mathcal{R}^+$, $\theta \in \mathcal{R}^{n_\theta}$, and assume that Φ is orthonormal, *i.e.,* $\Phi^T\Phi = \Phi\Phi^T = I$. Equation (4.40) can then be rewritten as

$$\min_{\theta} \left\|\Phi(\Phi^T y - \theta)\right\|_2^2 + \lambda\|\theta\|_1, \tag{4.41}$$

and since Φ can be seen as a rotation, which does not change the Euclidean length, of the vector $(\Phi^T y - \theta)$, $\left\|\Phi(\Phi^T y - \theta)\right\|_2^2 = \left\|\Phi^T y - \theta\right\|_2^2$. We can further rewrite $\|\Phi^T y - \theta\|_2^2$ using that $\Phi^T\Phi = I$ so that $\|\Phi^T y - \theta\|_2^2 = \left\|(\Phi^T\Phi)^{-1}\Phi^T y - \theta\right\|_2^2$. If we notice that $(\Phi^T\Phi)^{-1}\Phi^T y$ is the least squares solution *i.e.,*

$$\theta_{\text{ls}} = \arg\min_{\theta} \|y - \Phi\theta\|_2^2 = (\Phi^T\Phi)^{-1}\Phi^T y, \tag{4.42}$$

the solution of (4.40) can be written as

$$\min_{\theta} \|\theta_{\text{ls}} - \theta\|_2^2 + \lambda\|\theta\|_1 = \min_{\theta} \sum_{i=1}^{n_\theta} \Big(\theta_{\text{ls}}(i) - \theta(i)\Big)^2 + \lambda|\theta(i)|. \tag{4.43}$$

We can now consider the estimate of each of the elements of θ separately. Let us consider $\theta(i)$. Taking the derivative w.r.t. $\theta(i)$ of

$$\Big(\theta_{\text{ls}}(i) - \theta(i)\Big)^2 + \lambda|\theta(i)| \tag{4.44}$$

gives

$$-2\Big(\theta_{\text{ls}}(i) - \theta(i)\Big) + \lambda \operatorname{sign}\Big(\theta(i)\Big), \quad \theta(i) \neq 0. \tag{4.45}$$

We have to handle $\theta(i) = 0$ separately. Setting the derivative equal to zero and solving gives

$$\theta(i) = \begin{cases} \theta_{\text{ls}}(i) - \lambda/2 & \text{if } \theta_{\text{ls}}(i) - \lambda/2 > 0 \\ \theta_{\text{ls}}(i) + \lambda/2 & \text{if } \theta_{\text{ls}}(i) + \lambda/2 < 0 \end{cases} \tag{4.46}$$

or

$$\theta(i) = \text{sign}\Big(\theta_{\text{ls}}(i)\Big)\Big(|\theta_{\text{ls}}(i)| - \lambda/2\Big). \tag{4.47}$$

For $|\theta_{\text{ls}}(i)| < \lambda/2$, $\theta(i) = 0$. The $\theta(i)$ minimizing (4.44) is hence

$$\theta(i) = \text{sign}\Big(\theta_{\text{ls}}(i)\Big)\min\Big(0, |\theta_{\text{ls}}(i)| - \lambda/2\Big). \tag{4.48}$$

Note that (4.48) holds for $i = 1, \ldots, n_\theta$. The relation, for this special case, between the least squares estimate θ_{ls} and the estimate from lasso is visualized in Figure 4.4. We see that lasso shrinks the least squares estimate and if the least squares parameter estimate is close enough to zero, lasso gives a parameter estimate identical to zero.

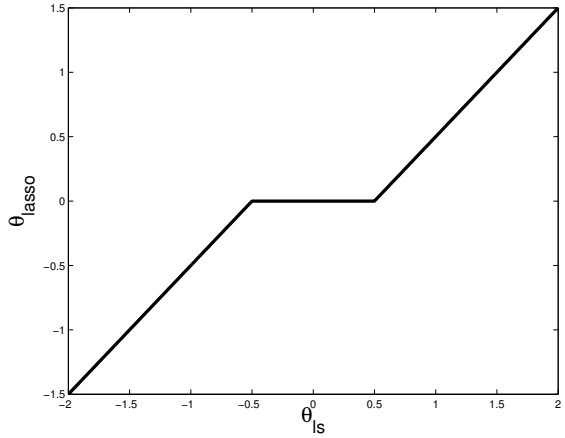

Figure 4.4: *The relation between the least squares estimate θ_{ls} and the estimate from lasso θ_{lasso} in the case where $\Phi^T\Phi = I$.*

4.3.2 Critical Parameter Value

Let us consider the ℓ_1-regularized least squares problem (4.20). A basic result from convex analysis tells us that there is a value $\lambda^{\max}$ for which the solution of the problem is equal to zero, if and only if $\lambda \geq \lambda^{\max}$. In other words, $\lambda^{\max}$ gives the threshold above which $\theta \equiv 0$. The critical parameter value $\lambda^{\max}$ is very useful in practice, since it gives a very good starting point in finding a suitable value of λ.

Proposition 4.1 (Critical Parameter Value $\lambda^{\max}$). *Let $\Phi \in \mathcal{R}^{N_e \times n}$ and $y \in \mathcal{R}^{N_e}$ be given. Let $\lambda^{\max}$ be such that θ minimizing*

$$\|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_1 \tag{4.49}$$

is zero if and only if $\lambda \geq \lambda^{\max}$. It holds that

$$\lambda^{\max} = \left\|2\Phi^T y\right\|_\infty. \tag{4.50}$$

The infinity-norm $\|\cdot\|_\infty$ is defined in Appendix A.

Proof: Define $\bar{e}_i$ as the n-dimensional row-vector with the ith element as one and the rest equal to zero. The subdifferential at $\theta = 0$ is readily computed to

$$\partial_{\theta(i)}\Big(\|y - \Phi\theta\|_2^2 + \lambda\|\theta\|_1\Big)\Big|_{\theta=0} = \Big[-2\bar{e}_i\Phi^T y - \lambda, -2\bar{e}_i\Phi^T y + \lambda\Big]. \tag{4.51}$$

For $\theta = 0$ to be an optima, it is necessary and sufficient (see *e.g.,* (Bertsekas et al., 2003, Prop. 4.7.2)) that

$$0 \in \Big[-2\bar{e}_i\Phi^T y - \lambda, -2\bar{e}_i\Phi^T y + \lambda\Big], \quad \forall i = 1, \ldots, n, \tag{4.52}$$

which is equivalent to

$$\lambda \geq \left\|2\Phi^T y\right\|_\infty. \tag{4.53}$$

(4.50) follows since $\lambda^{\max}$ is the smallest λ-value that makes $\theta = 0$ an optima. □

4.3.3 Sum-of-Norms Regularization

A ℓ_1-related regularization is the *sum-of-norms regularization*. A sum-of-norms regularized problem takes the form

$$\min_\theta V(\theta) + \lambda \sum_{i=1}^{s} \|\Gamma(i,:)\theta\|_p, \tag{4.54}$$

with $s \in \mathcal{N}$, Γ an $s \times dim(\varphi)\,(0,1)$-matrix and $\lambda \in \mathcal{R}^+$. The matrix Γ picks out groups of θ-elements. With $V(\theta) = \|y - \Phi\theta\|_2^2$ and $p = 2$ in (4.54),

$$\min_\theta \|y - \Phi\theta\|_2^2 + \lambda \sum_{i=1}^{s} \|\Gamma(i,:)\theta\|_2, \tag{4.55}$$

the formulation is often referred to as *group-lasso* (Yuan and Lin, 2006) in statistics. Note that the sum-of-norms regularization reduces to a ℓ_1-regularization if $\Gamma = I$ and $p = 1$ in (4.54).

We should comment on the difference between using an ℓ_1-regularization and some other type of sum-of-norms regularization, such as sum-of-Euclidean norms with $\Gamma \neq I$. When we use sum-of-norms regularization, the vector $\Gamma\theta$ will be sparse and when an element of the vector $\Gamma\theta$ is non-zero, say element i, then in general most of the θ-elements picked out by $\Gamma(i,:)$ are non-zero. The sum-of norms regularization hence makes sure that θ is sparse on a group-level, rather than an individual level.

Remark 4.3. Notice that (4.54) can be rewritten as

$$\min_{\theta} V(\theta) + \lambda \|\bar{\theta}\|_1, \quad \bar{\theta}(i) \triangleq \|\Gamma(i,:)\theta\|_p, \; i = 1, \ldots, s. \tag{4.56}$$

This clarifies the relation to the ℓ_1-regularization and provides an intuition for why groups of θ-elements ($\Gamma(i,:)\theta$, $i = 1, \ldots, s$) come out as zero or non-zero.

We continue the discussion on sum-of-norms regularization in Paper A, B, C and D.

4.3.4 Solution Methods

Many standard methods of convex optimization can be used to solve the problems (4.20) and (4.55). Software packages such as `CVX` (Grant and Boyd, 2010, 2008) or YALMIP (Löfberg, 2004) can readily handle the sum-of-norms regularization, by converting the problem to a cone problem and calling a standard interior-point cone solver. For the special case when the ℓ_1 norm is used as the regularization norm, more efficient special purpose algorithms and software can be used, such as `l1_ls` (Kim et al., 2007).

Recently many authors have developed fast first order methods for solving ℓ_1-regularized problems, and these methods can be extended to handle the sum-of-norms regularization, see *e.g.*, Roll (2008§2.2). Both interior-point and first-order methods have a complexity that scales linearly with N ($= dim(y)$ in (4.20)).

It has also been shown how solving ℓ_1-regularized problems can considerably be speeded up by pre-computing certain quantities (Mattingley and Boyd, 2010). It was shown how real-time performance can be met in many scenarios where ℓ_1-regularization previously was considered to be computationally too heavy.

CVX, YALMIP and l1_ls

CVX and YALMIP are very useful tools for solving ℓ_1 and sum-of-norms regularized (convex) problems. Both CVX and YALMIP are integrated with MATLAB. If we let

$$y = \begin{bmatrix} y_1 & y_2 & \ldots & y_{N_e} \end{bmatrix}^T, \; \Phi = \begin{bmatrix} \varphi_1 & \varphi_2 & \ldots & \varphi_{N_e} \end{bmatrix}^T, \; \Phi \in \mathcal{R}^{N_e \times n}, \; \lambda \in \mathcal{R}^+, \tag{4.57}$$

the ℓ_1-regularized least squares problem

$$\min_{\theta} \|y - \Phi\theta\|_2^2 + \lambda \|\theta\|_1 \tag{4.58}$$

can be solved using the CVX code given in Listing 4.1 and the YALMIP code given in Listing 4.2, assuming that the CVX respectively the YALMIP code package has been downloaded and installed. "y", "Phi", "n" and "lambda" also need to be available in the MATLAB workspace according to (4.57).

Listing 4.1: *CVX code for solving* (4.58)

```
cvx_begin
variable theta(n)
minimize((y-Phi*theta)'*(y-Phi*theta) ...
  +lambda*norm(theta,1))
cvx_end
```

Listing 4.2: *YALMIP code for solving* (4.58)

```
theta=sdpvar(n,1);
ops=sdpsettings('verbose',0);
solvesdp([],(y-Phi*theta)'*(y-Phi*theta) ...
   +lambda*norm(theta,1),ops)
```

A MATLAB package dedicated to ℓ_1-regularized least squares problems is l1_ls. With "y", "Phi" and "lambda" available in the MATLAB workspace according to (4.57) and the l1_ls package downloaded and installed, (4.58) can be solved as shown in Listings 4.3.

Listing 4.3: *l1_ls code for solving* (4.58)

```
rel_tol = 0.01; % relative target duality gap
theta=l1_ls(Phi,y,lambda,rel_tol)
```

4.4 Conclusion

This chapter has demonstrated how regularization can be used to obtain sparsity. There are a number of problems in system identification and signal processing that well fit into the framework developed. We therefore return to sparsity and regularization in Paper A, B, C and D.

5

Regularization for Smoothness

Regularization can be used to obtain meaningful results from ill-posed problems and to control for overfit. We care for both these applications in this thesis. However, we chose to focus on the type of regularization (referred to as a *standard regularization method* in Poggio et al. (1985)) obtained by adding a penalty term J to the criterion of fit,

$$\hat{\theta} = \arg\min_{\theta} \sum_{t \in \mathcal{N}_e} l\Big(y_t - f(\varphi_t, \theta)\Big) + \lambda J(\varphi_t, \theta), \quad \lambda \in \mathcal{R}^+. \tag{5.1}$$

The penalty J should be regarded as a means to introduce a *priori* knowledge and can be used to impose signal and model properties such as sparsity (discussed in Chapter 4) and smoothness. We discuss regularization for smoothness in this chapter. Geometrically, regularization for smoothness means that we seek the least rough function that gives a certain degree of fit to the observed data. Smoothness is in the regularization-literature used interchangeably with *curvature*, *non-rough*, *simplest* and *least complex*. The regularization parameter λ is used to control the trade-off between fit and smoothness.

Examples of regression methods that can be interpreted as regression methods that use regularization for smoothness are *support vector regression* and *Gaussian processes*. We give an introduction to these two methods in the following two sections.

5.1 Support Vector Regression

Let $\{(\varphi_t, y_t)\}_{t=1}^{N_e}$, $y_t \in \mathcal{R}$, $\varphi_t \in \mathcal{R}^{n_\varphi}$, be a given estimation data set and let $\{h_k(\cdot) : \mathcal{R}^{n_\varphi} \to \mathcal{R},\ k = 1, \ldots, n\}$ be a set of basis functions. It could *e.g.,* be the n first basis

functions of a Fourier series expansion. Consider now the task of estimating the basis function coefficients $\theta_k \in \mathcal{R}$, $k = 1, \ldots, n$, in the basis function expansion model

$$f(\varphi, \theta) = \sum_{k=1}^{n} h_k(\varphi)\theta_k. \tag{5.2}$$

Assume that $n > N_e$. Seeking the model parameters that minimize the sum of squared residuals

$$\sum_{t=1}^{N_e} \Big(y_t - \sum_{k=1}^{n} h_k(\varphi_t)\theta_k\Big)^2 \tag{5.3}$$

leads to an ill-posed problem since $n > N_e$ (see Example 2.2). In particular, the solution will generally not be unique. We saw previously, see Example 2.2, how ℓ_2-regularization can be used to transform (5.3) into a well-posed problem. If we introduce

$$y \triangleq \begin{bmatrix} y_1 & \ldots & y_{N_e} \end{bmatrix}^T, \; \theta \triangleq \begin{bmatrix} \theta_1 & \ldots & \theta_n \end{bmatrix}^T, \; h(\varphi_t) \triangleq \begin{bmatrix} h_1(\varphi_t) & \ldots & h_n(\varphi_t) \end{bmatrix}^T \tag{5.4}$$

and the matrix $H \in \mathcal{R}^{N_e \times n}$

$$H \triangleq \begin{bmatrix} h(\varphi_1) & \ldots & h(\varphi_{N_e}) \end{bmatrix}^T, \tag{5.5}$$

the ℓ_2-regularized least-squares criterion can be written as

$$\min_{\theta} \|y - H\theta\|^2 + \lambda\|\theta\|^2, \quad \lambda \in \mathcal{R}^+. \tag{5.6}$$

The minimizing θ is then readily computed as (see *e.g.,* (2.27))

$$\hat{\theta} = (H^T H + \lambda I_n)^{-1} H^T y. \tag{5.7}$$

Let now φ_* be a given new regressor. The basis function model (5.2) evaluated at φ_* takes the form

$$f(\varphi_*, \hat{\theta}) = h(\varphi_*)^T \hat{\theta} = h(\varphi_*)^T \Big(H^T H + \lambda I_n\Big)^{-1} H^T y \tag{5.8}$$

or equivalently

$$f(\varphi_*, \hat{\theta}) = h(\varphi_*)^T H^T \Big(HH^T + \lambda I_{N_e}\Big)^{-1} y. \tag{5.9}$$

We could be satisfied and stop here. The sought basis function coefficients are provided by (5.7) and (5.9) gives a formula for the basis function expansion evaluated at a new regressor φ_*. Let us continue and consider what happens when n gets very large. It can then become computationally impossible to evaluate (5.8) and (5.9). To be able to handle large n, define $k(\varphi_i, \varphi_j) : \mathcal{R}^{n_\varphi \times n_\varphi} \to \mathcal{R}$ as

$$k(\varphi_i, \varphi_j) \triangleq h(\varphi_i)^T h(\varphi_j). \tag{5.10}$$

(5.9) can then be rewritten as

$$f(\varphi_*, \hat{\theta}) = k(\varphi_*, \Phi)\Big(k(\Phi, \Phi) + \lambda I_{N_e}\Big)^{-1} y \tag{5.11}$$

where

$$\Phi = \begin{bmatrix} \varphi_1 & \dots & \varphi_{N_e} \end{bmatrix}^T, \tag{5.12}$$

$$k(\varphi_*, \Phi) = \begin{bmatrix} k(\varphi_*, \varphi_1) & \dots & k(\varphi_*, \varphi_{N_e}) \end{bmatrix}, \tag{5.13}$$

$$k(\Phi, \Phi) = \begin{bmatrix} k(\varphi_1, \varphi_1) & k(\varphi_1, \varphi_2) & \dots & k(\varphi_1, \varphi_{N_e}) \\ k(\varphi_2, \varphi_1) & k(\varphi_2, \varphi_2) & & k(\varphi_2, \varphi_{N_e}) \\ \vdots & & \ddots & \vdots \\ k(\varphi_{N_e}, \varphi_1) & k(\varphi_{N_e}, \varphi_2) & \dots & k(\varphi_{N_e}, \varphi_{N_e}) \end{bmatrix}. \tag{5.14}$$

In this way, we have avoided the basis functions h_k, $k = 1, \dots, n$, but anyway found a way to evaluate the model (5.2). Also when n is infinite the solution is given by (5.11), as shown by the *representer theorem* (see *e.g.,* Kimeldorf and Wahba (1971)). This is useful! This means that we can replace the computation of an infinite number of basis function coefficients with $N_e^2 + N_e$ evaluations of $k(\cdot, \cdot)$. One may wonder when it is possible to rewrite the dot-product $h(\varphi_i)^T h(\varphi_j)$ as in (5.10). And also, when is it possible to rewrite a function $k(\varphi_i, \varphi_j)$ as a dot-product between basis functions? In fact, in practice the function $k(\varphi_i, \varphi_j)$ is chosen and the particular form of the basis functions often not derived or thought of. To guarantee that $k(\varphi_i, \varphi_j)$ can be written as a dot-product between basis functions, $k(\varphi_i, \varphi_j)$ should be chosen as a symmetric, positive semi-definite kernel (see Mercer's theorem *e.g.,* Evgeniou et al. (2000) or Schölkopf and Smola (2001, p. 37), see also Appendix A). The squared exponential kernel has these properties (see Appendix A for definition and examples of more kernels).

The kernel can here be seen as a way to redefine the dot-product in the regressor space. This trick of redefining the dot-product can be used in regression methods where regressors only enter as products. These types of methods are surprisingly many and the usage of this trick results in the *kernelized*, or simply kernel, version of the method. (5.11) is a special case of *Least Squares Support Vector Machines regression* (LS-SVM regression or LS-SVR, see *e.g.,* Saunders et al. (1998); Suykens and Vandewalle (1999)).

By kernelizing a regression method, the regressor space is transformed by h to a possibly infinite dimensional new space in which the regression takes place. The transformation of the regression problem to a new high-dimensional space is commonly referred to as the *kernel trick* (Boser et al., 1992).

Example 5.1: Illustration of the Kernel Trick

Let $\varphi_1 = \begin{bmatrix} \varphi_1(1) & \varphi_1(2) \end{bmatrix}^T$, $\varphi_2 = \begin{bmatrix} \varphi_2(1) & \varphi_2(2) \end{bmatrix}^T$ and $\varphi_* = \begin{bmatrix} \varphi_*(1) & \varphi_*(2) \end{bmatrix}^T$ be three regressors in $\mathcal{R}^2$. Observe that if we use

$$k(\varphi_1, \varphi_2) = \varphi_1^T \varphi_2 = \varphi_1(1)\varphi_2(1) + \varphi_1(2)\varphi_2(2) \tag{5.15}$$

in (5.11) we get exactly the same expression as in (2.27) *i.e.,* ridge regression. Let us now use the kernel (polynomial (inhomogeneous) kernel, see Appendix A)

$$\tilde{k}(\varphi_1, \varphi_2) = (1 + \varphi_1^T \varphi_2)^2. \tag{5.16}$$

This could also be thought of as changing the definition of the dot-product between two regression vectors. We see that the regressors now affect the regression algorithm through

$$\begin{aligned}\tilde{k}(\varphi_1, \varphi_2) =& (1 + \varphi_1^T \varphi_2)^2 && \text{(5.17a)}\\ =& 1 + 2\varphi_1(1)\varphi_2(1) + 2\varphi_1(2)\varphi_2(2) + \varphi_1(1)^2\varphi_2(1)^2 \\ &+ \varphi_1(2)^2\varphi_2(2)^2 + 2\varphi_1(1)\varphi_1(2)\varphi_2(1)\varphi_2(2). && \text{(5.17b)}\end{aligned}$$

We can rewrite this as the dot-product between the vector valued function $h(\cdot)$ evaluated at φ_1 and φ_2

$$\tilde{k}(\varphi_1, \varphi_2) = h(\varphi_1)^T h(\varphi_2) \tag{5.18}$$

with

$$h(\varphi_1) = \begin{bmatrix} 1 & \sqrt{2}\varphi_1(1) & \sqrt{2}\varphi_1(2) & \varphi_1(1)^2 & \varphi_1(2)^2 & \sqrt{2}\varphi_1(1)\varphi_1(2) \end{bmatrix}^T \tag{5.19}$$

and $h(\varphi_2)$ accordingly. The polynomial (inhomogeneous) kernel hence transform the regressor space into a 6-dimensional space. If we now assume that an estimation data set $\{(\varphi_t, y_t)\}_{t=1}^{N_e}$ is given. Then in the particular case of LS-SVR, a linear model in $\mathcal{R}^6$ would be estimated to fit the transformed estimation data $\{(h(\varphi_t), y_t)\}_{t=1}^{N_e}$ using ridge regression. Reformulated in terms of the original regressors, the model evaluated at φ_* becomes

$$\begin{aligned}f(\varphi_*, \theta) =& \theta_1 + \sqrt{2}\theta_2\varphi_*(1) + \sqrt{2}\theta_3\varphi_*(2) + \theta_4\varphi_*(1)^2 + \theta_5\varphi_*(2)^2 \\ &+ \sqrt{2}\theta_6\varphi_*(1)\varphi_*(2). && \text{(5.20)}\end{aligned}$$

We see that by using this modified definition of the dot-product in LS-SVR we obtain a, in the regressors, polynomial predictor. We can hence compute nonlinear predictors by simply redefining the dot-product used in the regression algorithms.

We return to LS-SVR in Example 5.2.

5.2 Gaussian Process Regression

Consider the setup

$$y_t = f_0(\varphi_t) + e_t, \quad e_t \sim N(0, \sigma^2),\ \varphi_t \in \mathcal{R}^{n_\varphi},\ y_t \in \mathcal{R}. \tag{5.21}$$

Let $\{(\varphi_t, y_t)\}_{t=1}^{N_e}$ be a given estimation data set and consider the task of finding an estimate for f_0 at a regressor φ_*. In *Gaussian Process Regression* (GPR, see *e.g.,* Rasmussen and Williams (2005), also called *Kriging*, see *e.g.,* Matheron (1973)) the output $f_0(\varphi)$ is assumed to be a *stochastic process*, a *Gaussian Process* (GP). Any samples taken from a (zero-mean) Gaussian process are by definition related by a (zero-mean) Gaussian probability distribution. In particular, $f_0(\varphi_i)$ and $f_0(\varphi_j)$

will be related by

$$\begin{bmatrix} f_0(\varphi_i) \\ f_0(\varphi_j) \end{bmatrix} \sim N\left(0_{2\times 1}, \begin{bmatrix} k(\varphi_i,\varphi_i) & k(\varphi_i,\varphi_j) \\ k(\varphi_j,\varphi_i) & k(\varphi_j,\varphi_j) \end{bmatrix}\right) \tag{5.22}$$

for some kernel k. Let now $\Phi \in \mathcal{R}^{N_e \times n_\varphi}$ contain the estimation regressors

$$\Phi \triangleq \begin{bmatrix} \varphi_1 & \dots & \varphi_{N_e} \end{bmatrix}^T, \tag{5.23}$$

φ_* be a new regressor and let $k(\varphi_*, \Phi)$ and $k(\Phi, \Phi)$ be as in (5.13) and (5.14). Then, using (5.22) we have that

$$\begin{bmatrix} f_0(\varphi_1) & f_0(\varphi_2) & \dots & f_0(\varphi_{N_e}) & f_0(\varphi_*) \end{bmatrix}^T \sim N\left(0_{N_e+1\times 1}, \begin{bmatrix} k(\Phi,\Phi) & k(\varphi_*,\Phi)^T \\ k(\varphi_*,\Phi) & k(\varphi_*,\varphi_*) \end{bmatrix}\right).$$

If we let y denote the estimation outputs, $y \triangleq \begin{bmatrix} y_1 & \dots & y_{N_e} \end{bmatrix}^T$, then y and $f_0(\varphi_*)$ are related by

$$\begin{bmatrix} y^T & f_0(\varphi_*) \end{bmatrix}^T \sim N\left(0_{N_e+1\times 1}, \begin{bmatrix} k(\Phi,\Phi) + \sigma^2 I_{N_e} & k(\varphi_*,\Phi)^T \\ k(\varphi_*,\Phi) & k(\varphi_*,\varphi_*) \end{bmatrix}\right). \tag{5.24}$$

The predictive (or conditional) distribution for the stochastic variable $f_0(\varphi_*)$ given the estimation data can then be expressed as

$$\begin{aligned} p\Big(f_0(\varphi_*)\Big|\{(\varphi_t, y_t)\}_{t=1}^{N_e}\Big) = N\Big(&k(\varphi_*,\Phi)(k(\Phi,\Phi) + \sigma^2 I_{N_e})^{-1} y, \\ &k(\varphi_*,\varphi_*) - k(\varphi_*,\Phi)(k(\Phi,\Phi) + \sigma^2 I_{N_e})^{-1} k(\varphi_*,\Phi)^T\Big) \end{aligned} \tag{5.25}$$

using identities for Gaussian distributions, see *e.g.,* (Rasmussen and Williams, 2005, p. 200). Notice that the (5.25) gives the distribution for the value of $f_0(\varphi_*)$ and not a measurement of f_0 at φ_*. To get the distribution for a measurement of f_0 at φ_*, σ^2 should be added to the covariance in (5.25). The kernel k defines the correlation between $f_0(\varphi_i)$ and $f_0(\varphi_j)$. This correlation is most often unknown and seen as a design choice in GPR. A popular choice is the squared exponential kernel, see Appendix A.

The predictive mean (mean of the distribution in (5.25)) takes exactly the same form as the prediction in least squares support vector regression, see (5.11). Gaussian process regression can hence also be given an interpretation as a regularization method.

Example 5.2: Gaussian Processes Regression (and LS-SVR)

Let $\{(\varphi_t, y_t)\}_{t=1}^{N_e}$, $N_e = 10$, be generated by

$$y_t = 5\sin\varphi_t + e_t, \quad e_t \sim N(0,1),\ \varphi_t \sim U(0,5). \tag{5.26}$$

The estimation data are shown with '+'-marks in Figure 5.1. If Gaussian process regression with k as a scaled squared exponential kernel

$$k(\varphi_i,\varphi_j) = \gamma^2 e^{-\|\varphi_i-\varphi_j\|_2^2/2\ell^2}, \tag{5.27}$$

with a length scale $\ell = 1$, $\gamma = 5$ and noise standard deviations $\sigma = 0.1,\ 1,\ 5$ and 20 are used, the predictive distributions (for noisy measurements of f_0) visualized in Figure 5.1 are obtained.

The predictive mean (mean of the distribution in (5.25)) takes exactly the same form as the prediction in least squares support vector regression, see (5.11). Hence, the solid line in Figure 5.1 could equally well have been the result from LS-SVR with the kernel (5.27) and $\lambda = \sigma^2$. As seen in Figure 5.1, σ^2, or the regularization parameter, controls the smoothness of the predictive mean. If we let σ^2 go to infinity, the function-estimate will approach zero and a very smooth function. If we instead let σ^2 go to zero, the function estimate will become more and more non-smooth. This behavior is rather intuitive since σ^2 has an interpretation as the measurement noise covariance.

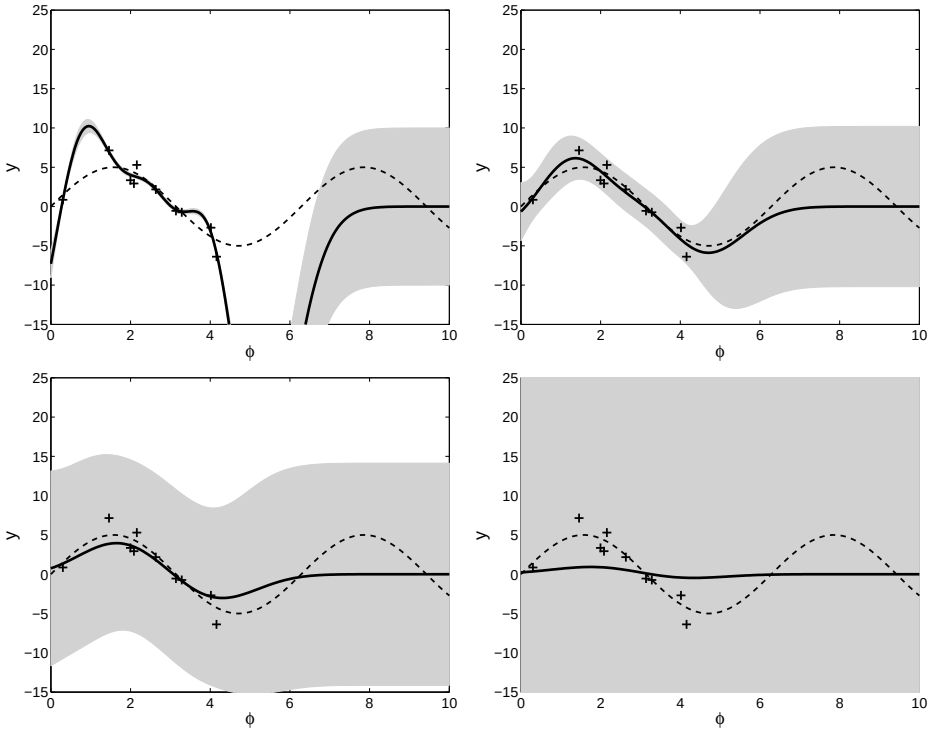

Figure 5.1: *Posterior (or predictive) distributions for a Gaussian process with $\ell = 1$, $\gamma = 5$ and $\sigma = 0.1$ (left top plot), 1 (right top plot), 5 (left bottom plot) and 20 (right bottom plot). The estimation data are shown with '+'-marks, the dashed line shows $5\sin(\cdot)$ and the solid line shows the mean of the predictive distribution or the LS-SVR estimate. The gray area shows the two standard deviations confidence interval for noisy measurements of f_0.*

Smoothness of the mean of the predictive distribution (5.25) is highly dependent on σ^2 (the regularization parameter). Parameters, such as σ and possible parameters of the kernel, that have to be set, are denoted *hyperparameters* (see Section 2.9 for hyperparmeters). The hyperparameters could be chosen using cross validation, but if few observations are available, maximizing the marginal likelihood is a good alternative (see Section 2.9).

Example 5.3: Gaussian Processes Regression Cont'd

Let us return to Example 5.2 and find the hyperparameters ℓ and γ of the scaled squared exponential kernel

$$k(\varphi_i, \varphi_j) = \gamma^2 e^{-\|\varphi_i - \varphi_j\|_2^2 / 2\ell^2} \tag{5.28}$$

and the measurement noise variance σ^2 by maximizing the marginal likelihood. The parameters were estimated to

$$\ell = 1.4,\ \gamma = 4.2,\ \sigma = 1.6, \tag{5.29}$$

using `GPML` (Rasmussen and Nickisch, 2010). `GPML` is a MATLAB toolbox for GPR. The code for estimating the hyperparameters using `GPML` are given in Listing 5.1.

Listing 5.1: *Estimation of hyperparameters ℓ, γ and σ using `GPML`.*

```
covfunc={'covSum',{'covSEard','covNoise'}};
loghyper=minimize([-1;-1;-1],'gpr',-100,covfunc,Phi,y);
[l gamma sigma]=exp(loghyper);
```

The resulting predictive distribution for noisy measurements of f_0 is visualized in Figure 5.2.

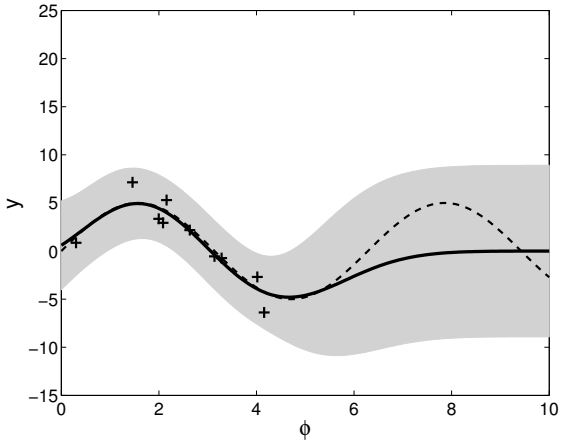

Figure 5.2: *The predictive distribution for noisy measurements of f_0. Mean given as a solid line and the gray area shows the two standard deviations confidence interval. The '+'-marks show the estimation data and the dashed line* $5\sin(\cdot)$.

5.3 Conclusion

Regularization for smoothness is essential in the estimation of many nonparametric models to obtain smooth estimates and control for overfitting. We have seen how both least squares support vector machines and Gaussian processes regression use regularization and how the regularization controlled the smoothness of the estimated model. We will continue the discussion on regularization and smoothness in Paper E and derive a novel regularization method, *Weight Determination by Manifold Regularization* (WDMR). Also Paper F discusses regularization for smoothness and in particular how it can be used to estimate impulse responses.

6

Concluding Remarks

6.1 Conclusion

The introductory part of this thesis was aimed to motivate and give a background to the papers of Part II. The focus was regularization and in particular, regularization for sparseness and smoothness. A number of examples of previous usages of regularization for sparseness and smoothness was given along with illustrative applications.

Part II of this thesis consists of a collection of papers. The first four papers utilize regularization for sparseness. First out is a novel optimization formulation for the identification of segmented ARX models, Paper A. Regularization for sparsity is there applied to control for overfitting. Paper B provides a novel system identification approach to piecewise affine systems. Regularization for sparsity is utilized to control for overfitting. Paper C discusses state estimation and provides a novel nonlinear smoother. The smoother works under the assumption that the process noise is impulsive, that is, often zero but occasionally nonzero. Regularization for sparsity again plays an important role to control for overfitting. The theory presented in this paper could be suitable in *e.g.,* target tracking applications. Paper D presents a novel model-based approach to trajectory generation. Regularization for sparsity is here used to find trajectories with compact representations. Paper E discusses regularization for smoothness. A novel regularization method *Weight Determination by Manifold Regularization* (WDMR) is presented. WDMR is inspired by manifold learning and applications in biology and has inherited properties thereof. WDMR uses regularization for smoothness to obtain smooth estimates. Paper F applies regularization for smoothness to linear system identification. In particular, high order FIR models are studied. Last, Paper G presents a real-time fMRI bio-feedback setup. The setup has served as a

proof of concept and shows that useful information can be read out, in real-time, from the brain activity measurements.

6.2 Future Research

It would be interesting to look at some more theoretical questions concerning the regularization methods and techniques developed in this thesis. A rather extensive theory has been developed around compressed sensing. This theory is not directly applicable to the methods presented in the papers of Part II on regularization for sparsity. It however provides tools for developing a deeper theoretical understanding. Interesting theoretical questions are:

- Under what assumptions can the correct sparsity pattern be found?
- How sensitive are the methods using regularization for sparsity for measurement noise? For example, how sensitive are the segmentation algorithm presented in Paper A to measurement noise?
- What happens if the number of estimation data samples goes to infinity? What is the asymptotic behavior?

There are also several possible application areas for regularization for sparseness which have not been explored. Multi-target tracking and event based sampling and control may for example be interesting areas for further research using regularization for sparseness.

It would also be interesting to investigate what techniques, such as, *General Principal Component Analysis* (GPCA, Vidal et al. (2003a,b, 2005)) can do for system identification and signal processing. GPCA has relations to sparsity techniques and has *e.g.,* been used in the identification of segmented ARX models, see *e.g.,* Vidal et al. (2003b). In particular, GPCA can be used to ensure that at least one element of a quantity is zero.

Interesting is also the development of new techniques and theories in machine learning. Many machine learning techniques are not directly applicable to dynamic systems, but they give a suitable foundation for the development of algorithms for dynamic systems. WDMR, presented in Paper E, is one example of such development. WDMR has shown useful in several applications, and there are for sure many interesting suitable applications as well as theoretical findings to be explored.

The last paper of this thesis, Paper G, discusses a real-time fMRI biofeedback setup. The potential of real-time fMRI is very exciting and applications of fMRI biofeedback have recently attract quite some attention in media and literature. It has *e.g.,* been shown how subjects can be trained to control their own pain using fMRI biofeedback (DeCharms et al., 2005). Our setup has been used as a communication interface (Eklund et al., 2010) and for real-time brain activity visualization (Nguyen et al., 2010). Many exciting applications remain to be explored, however.

6.3 Further Readings

For readers familiar with system identification that would like to read more about the mathematical background on underdetermined systems, sparseness and regularization, a very nice reading is Bruckstein et al. (2009). The paper by Zibulevsky and Elad (2010) also gives a nice introduction to sparsity. For a nice book that discusses several different regularization methods, Hastie et al. (2001) is to recommend. For the reader interested in machine learning and Bayesian modeling, Bishop (2006) is a good reference. Gaussian processes are nicely presented in Rasmussen and Williams (2005).

A

Kernels and Norms

This appendix lists a number of kernels and norms used in this thesis. Some properties of kernels are also discussed.

A.1 Kernels

In machine learning, a *kernel* $k : \mathcal{X} \times \mathcal{X} \to \mathcal{R}$ is a general name for a function of two arguments mapping to $\mathcal{R}$. A kernel is said to be *symmetric* (see *e.g.,* Rasmussen and Williams (2005, p. 80)) if

$$k(\varphi_i, \varphi_j) = k(\varphi_j, \varphi_i), \tag{A.1}$$

for any two $\varphi_i, \varphi_j \in \mathcal{X}$. If the kernel is going to be used in GPR as a covariance function, it needs to be symmetric. A kernel is said to be *stationary* (see *e.g.,* Rasmussen and Williams (2005, p. 79)) if $k(\varphi_i, \varphi_j)$ can be written as

$$k(\varphi_i, \varphi_j) = \bar{k}(\varphi_i - \varphi_j), \quad \varphi_i, \varphi_j \in \mathcal{X}, \tag{A.2}$$

for some function $\bar{k} : \mathcal{X} \to \mathcal{R}$. It is *non-stationary* if not stationary. Last, a kernel is said to be *positive semi-definite* (see *e.g.,* Rasmussen and Williams (2005, p. 80)) if for any number of inputs $\varphi_1, \ldots, \varphi_N$ in $\mathcal{X}$, the *Gram matrix* K with element ij given by $k(\varphi_i, \varphi_j)$ is positive semi-definite.

A symmetric positive semi-definite kernel k can be written as a dot-product

$$k(\varphi_i, \varphi_j) = h^T(\varphi_i)h(\varphi_j), \quad \varphi_i, \varphi_j \in \mathcal{X}. \tag{A.3}$$

This follows from Mercer's theorem (see *e.g.,* Schölkopf and Smola (2001, pp. 37-38)). $h(\cdot)$ is called a *feature map*.

See Rasmussen and Williams (2005, Chap. 4) or Schölkopf and Smola (2001, Chap. 2) for further discussions on kernels and their properties.

Remark A.1. The precise mathematical definition of a kernel states that a kernel is a function $k : \mathcal{X} \times \mathcal{X} \to \mathcal{R}$ that is both symmetric and positive semi-definite. We use the more liberal definition of machine learning.

A.1.1 Squared Exponential Kernel

For two vectors $\varphi_i, \varphi_j \in \mathcal{R}^n$, define the *squared exponential kernel* (sometimes called a *Gaussian kernel* or *Gaussian radial basis kernel*) as

$$k(\varphi_i, \varphi_j) \triangleq e^{-\|\varphi_i - \varphi_j\|_2^2 / 2\ell^2}, \tag{A.4}$$

where ℓ is a parameter of the kernel and denoted the *length scale*. The squared exponential kernel is symmetric, stationary and positive definite (Micchelli, 1986).

A.1.2 Polynomial Kernel

For two vectors $\varphi_i, \varphi_j \in \mathcal{R}^n$, define the *polynomial (inhomogeneous) kernel* as

$$k(\varphi_i, \varphi_j) \triangleq (\varphi_i^T \varphi_j + 1)^d, \quad d \in \mathcal{N}. \tag{A.5}$$

The *feature map*, or h, associated with the polynomial (inhomogeneous) kernel contains all monomials of order up to d (*e.g.,* Schölkopf et al. (2001, Prop. 2.17)). The polynomial kernel is symmetric, non-stationary and positive definite (see *e.g.,* Vapnik (1995, p. 460)).

A.2 Norms

A.2.1 Infinity Norm

For a vector $x \in \mathcal{R}^n$, define the infinity-norm as

$$\|x\|_\infty \triangleq \max_{i=1,\ldots,n} |x(i)|. \tag{A.6}$$

A.2.2 ℓ_0-Norm

For a vector $x \in \mathcal{R}^n$, define the zero (quasi-)norm as

$$\|x\|_0 \triangleq card\Big(\Big\{i \Big| x(i) \neq 0\Big\}\Big). \tag{A.7}$$

The zero norm is the number of nonzero elements of the vector x. The zero norm is a quasi-norm since it is not *positive homogeneous*. That is, the zero norm does not satisfy

$$\|\alpha x\|_0 \neq |\alpha| \|x\|_0, \quad \alpha \in \mathcal{R}, \tag{A.8}$$

which all norms should.

A.2.3 ℓ_p-Norm ($0 < p < \infty$)

For a vector $x \in \mathcal{R}^n$, define the ℓ_p-norm, $0 < p < \infty$, as

$$\|x\|_p \triangleq \left(\sum_{i=1}^{n} |x(i)|^p \right)^{1/p}. \tag{A.9}$$

The ℓ_2-norm is referred to as the *Euclidean norm*. See Figure 4.1, p. 52, for a visualization of some different ℓ_p-norms.

B

Huber Cost Function as a ℓ_1-Regularized Least Squares Problem

We use this appendix to show that the ℓ_1-regularized least squares formulation

$$\min_{\theta,\eta_1,\dots,\eta_N} \sum_{t=1}^{N_e} (y_t - \varphi_t^T\theta - \eta_t)^2 + \lambda \left\| \begin{bmatrix} \eta_1 & \eta_2 & \dots & \eta_{Ne} \end{bmatrix} \right\|_1. \tag{B.1}$$

derived in Examples 4.3 and 4.6 is minimized by the same θ as

$$\min_{\theta} \sum_{t=1}^{N_e} \psi\big(y_t - \varphi_t^T\theta\big) \tag{B.2}$$

with

$$\psi(x) \triangleq \begin{cases} |x|^2, & \text{if } |x| < \lambda/2, \\ \lambda|x| - \lambda^2/4 & \text{otherwise.} \end{cases} \tag{B.3}$$

First notice that (B.1) is equivalent to

$$\min_{\theta,\eta_1,\dots,\eta_N} \sum_{t=1}^{N_e} \big((y_t - \varphi_t^T\theta - \eta_t)^2 + \lambda|\eta_t|\big). \tag{B.4}$$

We now aim to show that

$$\min_{\eta_t}(y_t - \varphi_t^T\theta - \eta_t)^2 + \lambda|\eta_t| = \psi(y_t - \varphi_t^T\theta). \tag{B.5}$$

Let us consider the left hand side of (B.5) and step-by-step derive the right hand side. First, notice that $|\eta_t| = \operatorname{sign}(\eta_t)\eta_t$ and

$$\frac{d}{d\eta_t}|\eta_t| = \frac{d}{d\eta_t}\operatorname{sign}(\eta_t)\eta_t = 2\delta(\eta_t)\eta_t + \operatorname{sign}(\eta_t), \tag{B.6}$$

the function $\delta(\cdot)$ denoting the Dirac delta function. Then

$$\frac{d}{d\eta_t}\left((y_t - \varphi_t^T\theta - \eta_t)^2 + \lambda|\eta_t|\right) = -2(y_t - \varphi_t^T\theta - \eta_t) + 2\lambda\delta(\eta_t)\eta_t + \lambda\,\mathrm{sign}(\eta_t).$$

Equating the derivative to zero and solve for η_t gives

$$\eta_t^* = y_t - \varphi_t^T\theta - \lambda\delta(\eta_t^*)\eta_t^* - \lambda/2\,\mathrm{sign}(\eta_t^*), \tag{B.7}$$

which is implicit in η_t^*. For a $\eta_t^* > 0$, (B.7) reduces to

$$\eta_t^* = y_t - \varphi_t^T\theta - \lambda/2 \tag{B.8}$$

which implies that $y_t - \varphi_t^T\theta > \lambda/2$. Equivalent, a $\eta_t^* < 0$ implies that $\eta_t^* = y_t - \varphi_t^T\theta + \lambda/2$ and $y_t - \varphi_t^T\theta < -\lambda/2$. Now, if $\lambda/2 \geq y_t - \varphi_t^T\theta \geq 0$, then $\eta_t \geq 0$, since otherwise it dose not counteract on the positive $y_t - \varphi_t^T\theta$ in the left hand side of (B.5). Using this, the left hand side of (B.5) becomes

$$\min_{\eta_t:\eta_t\geq 0}(y_t - \varphi_t^T\theta - \eta_t)^2 + \lambda\eta_t = \min_{\eta_t:\eta_t\geq 0}\eta_t(\eta_t + 2(\lambda/2 - (y_t - \varphi_t^T\theta))). \tag{B.9}$$

Since $\lambda/2 - (y_t - \varphi_t^T\theta) \geq 0$, $\eta_t^* = 0$ minimizes (B.9). Similarly, if $-\lambda/2 \leq y_t - \varphi_t^T\theta \leq 0$, then $\eta_t \leq 0$ which leads to

$$\min_{\eta_t:\eta_t\leq 0}(y_t - \varphi_t^T\theta - \eta_t)^2 - \lambda\eta_t = \min_{\eta_t:\eta_t\leq 0}\eta_t(\eta_t - 2(\lambda/2 + y_t - \varphi_t^T\theta)) \tag{B.10}$$

and again the same solution, $\eta_t^* = 0$. All together

$$\eta_t^* = \begin{cases} y_t - \varphi_t^T\theta - \lambda/2, & y_t - \varphi_t^T\theta > \lambda/2, \\ 0, & |y_t - \varphi_t^T\theta| < \lambda/2, \\ y_t - \varphi_t^T\theta + \lambda/2, & y_t - \varphi_t^T\theta < -\lambda/2. \end{cases} \tag{B.11}$$

(B.11) inserted in $(y_t - \varphi_t^T\theta - \eta_t)^2 + \lambda|\eta_t|$ gives

$$\begin{aligned}
&\min_{\eta_t}(y_t - \varphi_t^T\theta - \eta_t)^2 + \lambda|\eta_t| && \text{(B.12a)}\\
&=(y_t - \varphi_t^T\theta - \eta_t^*)^2 + \lambda|\eta_t^*| && \text{(B.12b)}\\
&=\begin{cases} \lambda^2/4 + \lambda|y_t - \varphi_t^T\theta - \lambda/2|, & \text{if } y_t - \varphi_t^T\theta > \lambda/2 \\ (y_t - \varphi_t^T\theta)^2, & |y_t - \varphi_t^T\theta| < \lambda/2 \\ \lambda^2/4 + \lambda|y_t - \varphi_t^T\theta + \lambda/2|, & y_t - \varphi_t^T\theta < -\lambda/2 \end{cases} && \text{(B.12c)}\\
&=\begin{cases} \lambda(y_t - \varphi_t^T\theta) - \lambda^2/4, & \text{if } y_t - \varphi_t^T\theta > \lambda/2 \\ (y_t - \varphi_t^T\theta)^2, & |y_t - \varphi_t^T\theta| < \lambda/2 \\ -\lambda(y_t - \varphi_t^T\theta) - \lambda^2/4, & y_t - \varphi_t^T\theta < -\lambda/2 \end{cases} && \text{(B.12d)}\\
&=\psi\left(y_t - \varphi_t^T\theta\right) && \text{(B.12e)}
\end{aligned}$$

where the last equality holds from the definition (B.3) of the Huber loss function. Since (B.5) holds for any θ, it follows that θ minimizing (B.1) also minimizes (B.2).

Bibliography

H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Proceedings of the 2nd International Symposium on Information Theory*, pages 267–281, Akademiai Kiado, Budapest, 1973.

B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, N.J., 1979.

K. J. Åström and P. Eykhoff. System identification – A survey. *Automatica*, 7: 123–162, 1971.

M. Bauwens, H. Ohlsson, K. Barbé, V. Beelaerts, F. Dehairs, and J. Schoukens. On climate reconstruction using bivalve shells: Three methods to interpret the chemical signature of a shell. In *Proceedings of the 7th IFAC Symposium on Modelling and Control in Biomedical Systems*, Aalborg, Denmark, August 2009a.

M. Bauwens, H. Ohlsson, K. Barbé, V. Beelaerts, J. Schoukens, and F. Dehairs. Three ways to do temperature reconstruction based on bivalve-proxy information. In *Proceedings of the 28th Benelux Meeting on Systems and Control*, Spa, Belgium, March 2009b.

M. Bauwens, H. Ohlsson, K. Barbé, V. Beelaerts, F. Dehairs, and J. Schoukens. On climate reconstruction using bivalves: Three methods to interpret the chemical signature of a shell. *Computer Methods and Programs in Biomedicine*, 2010a. Accepted for publication.

M. Bauwens, H. Ohlsson, K. Barbé, V. Beelaerts, F. Dehairs, and J. Schoukens. A nonlinear multi-proxy model based on manifold learning to reconstruct water temperature from high resolution trace element profiles in biogenic carbonates. *Geoscientific Model Development*, 2010b. Accepted for publication.

T. Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1763.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

Y. Bengio, O. Delalleau, and N. Le Roux. The curse of highly variable functions for local kernel machines. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18 of *Neural Information Processing*. MIT Press, 2006.

D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.

J. M. Bioucas-Dias. Bayesian wavelet-based image deconvolution: A GEM algorithm exploiting a class of heavy-tailed priors. *IEEE Transactions on Image Processing*, 15(4):937–951, April 2006.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, August 1988.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th annual workshop on Computational learning theory (COLT'92)*, pages 144–152, New York, NY, USA, 1992. ACM.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

S. P. Brady, M. N. Do, and R. Bhargava. Reconstructing FT-IR spectroscopic imaging data with a sparse prior. In *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP)*, pages 829–832, November 2009.

K. Brandenburg. MP3 and AAC explained. In *Proceedings of the Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding*, Florence, Italy, September 1999.

A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1): 34–81, 2009.

E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, March 2008.

E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52:489–509, February 2006.

E. J. Candès, Y. C. Eldar, and D. Needell. Compressed sensing with coherent and redundant dictionaries. *CoRR*, abs/1005.2613, 2010.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

B. Chen, J. Paisley, and L. Carin. Sparse linear regression with beta process priors. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 1234–1237, Dallas, TX, March 2009.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

T. Chen, T. B. Schön, H. Ohlsson, and L. Ljung. Decentralization of particle filters using arbitrary state partitioning. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, USA, December 2010a. To appear.

T. Chen, T. B. Schön, H. Ohlsson, and L. Ljung. Decentralized particle filter with arbitrary state partitioning. *IEEE Transactions on Signal Processing*, 2010b. Accepted for publication.

T. Chen, H. Ohlsson, and L. Ljung. On the estimation of transfer functions, regularizations and Gaussian processes – Revisited. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011. Submitted.

C. Daniel and F. S. Wood. *Fitting Equations to Data: Computer Analysis of Multifactor Data*. John Wiley & Sons, Inc., New York, NY, USA, 1980.

D. de Ridder and R. P.W. Duin. Locally linear embedding for classification, 2002. Technical Report, PH-2002-01, Pattern Recognition Group, Dept. of Imaging Science & Technology, Delft University of Technology, Delft, The Netherlands.

R. C. DeCharms, F. Maeda, G. H. Glover, D. Ludlow, J. M. Pauly, S. Whitfield, J. D. E. Gabrieli, and S. C. Mackey. Control over brain activation and pain learned by using real-time functional MRI. *Proc Natl Acad Sci USA*, 102:18626–18631, 2005.

M. P. Deisenroth and H. Ohlsson. General perspective to Gaussian filtering and smoothing: Explaining current and deriving new algorithms. In *Proceedings of the American Control Conference (ACC), 2011*, San Francisco, USA, 2011. Submitted.

D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.

D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences of the United States of America*, 100(10):5591–5596, 2003.

B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

A. Eklund, H. Ohlsson, M. Andersson, J. Rydell, A. Ynnerman, and H. Knutsson. Using real-time fMRI to control a dynamical system. In *Proceedings of the 17th Meeting of the International Society for Magnetic Resonance in Medicine (ISMRM)*, Honolulu, USA, April 2009a.

A. Eklund, H. Ohlsson, M. Andersson, J. Rydell, A. Ynnerman, and H. Knutsson. Using real-time fMRI to control a dynamical system by brain activity classification. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'09)*, London, UK, September 2009b.

A. Eklund, M. Andersson, H. Ohlsson, A. Ynnerman, and H. Knutsson. A brain computer interface for communication using real-time fMRI. In *Proceedings of the International Conference on Pattern Recognition 2010*, Istanbul, Turkey, August 2010.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1–50, 2000.

T. Falck, H. Ohlsson, L. Ljung, J. A.K. Suykens, and B. De Moor. Segmentation of times series from nonlinear dynamical systems. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011. Submitted.

M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. D. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. `http://cvxr.com/cvx`, August 2010.

Z. Guo, C. Li, L. Song, and Wang L. V. Compressed sensing in photoacoustic tomography in vivo. *Journal of Biomedical Optics*, 15(2), 2010.

F. Gustafsson. *Adaptive Filtering and Change Detection*. Wiley, New York, 2001.

F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur AB, 2010.

J. Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, 13:49–52, 1902.

T. Hastie, R Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

B. Hayes. The best bits. *American Scientist*, 97(4):276–280, 2009.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990. Cambridge.

J. Hu, J. Tian, and L. Yang. Functional feature embedded space mapping of fMRI data. In *Proceedings of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS'06)*, pages 1014–1017, 2006.

P. J. Huber. Robust regression: Asymptotics, conjectures and Monte Carlo. *The Annals of Statistics*, 1(5):799–821, 1973.

X. Huo and X. Ni. When do stepwise algorithms meet subset selection criteria? *Annals of Statistics*, 35(2):870–887, August 2007.

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice-Hall, Englewood Cliffs, NJ, 2000.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):606–617, December 2007.

G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(82–95), 1971.

J. M. Lee. *Introduction to Topological Manifolds (Graduate Texts in Mathematics)*. Springer, May 2000.

F. Lindsten, J. Callmer, H. Ohlsson, D. Törnqvist, T. B. Schön, and F. Gustafsson. Geo-referencing for UAV navigation using environmental classification. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, May 2010.

L. Ljung. *System Identification - Theory for the User*. Prentice-Hall, Upper Saddle River, N.J., 2nd edition, 1999.

L. Ljung. Prediction error estimation methods. *Circuits, Systems, and Signal Processing*, 21:11–21, 2002.

L. Ljung and T. Kailath. A unified approach to smoothing formulas. *Automatica*, 12(2):147–157, 1976.

J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. URL `http://control.ee.ethz.ch/~joloef/yalmip.php`.

S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, December 1993.

G. Matheron. The intrinsic random functions and their applications. *Advances in Applied Probability*, 5(3):439–468, 1973.

J. Mattingley and S. Boyd. Real-time convex optimization in signal processing. *IEEE Signal Processing Magazine*, 27(3):50–61, 2010.

C. A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986.

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

A. Neumaier. Solving ill-conditioned and singular linear systems: A tutorial on regularization. *SIAM Rev.*, 40(3):636–666, 1998.

K. Nguyen, A. Eklund, H. Ohlsson, F. Hernell, P. Ljung, C. Forsell, M. Andersson, H. Knutsson, and A. Ynnerman. Concurrent volume visualization of real-time fMRI. In *Proceedings of the IEEE International Symposium on Volume Graphics 2010*, Norrköping, Sweden, May 2010.

H. Ohlsson. *Regression on manifolds with implications for system identification*. Licentiate thesis no. 1382, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, December 2008.

H. Ohlsson and L. Ljung. Gray-box identification for high-dimensional manifold constrained regression. In *Proceedings of the 15th IFAC Symposium on System Identification, SYSID 2009*, Saint-Malo France, July 2009.

H. Ohlsson and L. Ljung. Semi-supervised regression and system identification. In X. Hu, U. Jonsson, B. Wahlberg, and B. Ghosh, editors, *Three Decades of Progress in Control Sciences*. Springer Verlag, December 2010a. To appear.

H. Ohlsson and L. Ljung. Weight determination by manifold regularization. In *Distributed Decision-Making and Control*, Lecture Notes in Control and Information Sciences. Springer Verlag, 2010b. Submitted.

H. Ohlsson and L. Ljung. Piecewise affine system identification using sum-of-norms regularization. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011. Submitted.

H. Ohlsson, J. Roll, T. Glad, and L. Ljung. Using manifold learning for nonlinear system identification. In *Proceedings of the 7th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, Pretoria, South Africa, August 2007.

H. Ohlsson, J. Roll, A. Brun, H. Knutsson, M. Andersson, and L. Ljung. Direct weight optimization applied to discontinuous functions. In *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008a.

H. Ohlsson, J. Roll, and L. Ljung. Manifold-constrained regressors in system identification. In *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008b.

H. Ohlsson, J. Rydell, A. Brun, J. Roll, M. Andersson, A. Ynnerman, and H. Knutsson. Enabling bio-feedback using real-time fMRI. In *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008c.

H. Ohlsson, M. Bauwens, and L. Ljung. On manifolds, climate reconstruction and bivalve shells. In *Proceedings of the 48th IEEE Conference on Decision and Control*, Shanghai, China, December 2009.

H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. State smoothing by sum-of-norms regularization. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, USA, December 2010a. To appear.

H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. Trajectory generation using sum-of-norms regularization. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, USA, December 2010b. To appear.

H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd. State estimation under abrupt changes using sum-of-norms regularization. *Automatica*, 2010c. Submitted, under revision.

H. Ohlsson, L. Ljung, and S. Boyd. Segmentation of ARX-models using sum-of-norms regularization. *Automatica*, 46(6):1107–1111, 2010d.

A. V. Oppenheim, A. S. Willsky, and S. H. Nawab. *Signals & Systems (2nd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

G. Pillonetto and G. De Nicolao. A new kernel-based approach for linear system identification. *Automatica*, 46(1):81–93, January 2010.

T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(26):314–319, 1985.

C. E. Rasmussen and H. Nickisch. GPML Gaussian processes for machine learning toolbox, 2010. Version 2.0, `http://www.gaussianprocess.org/gpml/code`.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, December 2005.

S. Riezler and A. Vasserman. Incremental feature selection and l1 regularization for relaxed maximum-entropy modeling. In D. Lin and D. Wu, editors, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 174–181, Barcelona, Spain, July 2004. Association for Computational Linguistics.

J. Roll. Piecewise linear solution paths with application to direct weight optimization. *Automatica*, 44:2745–2753, 2008.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the 15th International Conference on Machine Learning*, pages 515–521. Morgan Kaufmann, 1998.

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory (COLT '01/EuroCOLT '01)*, pages 416–426, London, UK, 2001. Springer-Verlag.

X. Shen and F. G. Meyer. *Analysis of Event-Related fMRI Data Using Diffusion Maps*, volume 3565/2005 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, July 2005.

J.-L. Starck, E. J. Candes, and D. L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):670–684, June 2002.

J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

B. Thirion and O. Faugeras. Nonlinear dimension reduction of fMRI data: The Laplacian embedding approach. *IEEE International Symposium on Biomedical Imaging: Nano to Macro*, 1:372–375, 2004.

R. Tibsharani. Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society B (Methodological)*, 58(1):267–288, 1996.

A.-I N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977.

J. A. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, March 2006.

J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk. Beyond Nyquist: Efficient sampling of sparse bandlimited signals. *IEEE Transactions on Information Theory*, 56(1):520–544, January 2010.

J.A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, October 2004.

V. Vapnik. *Estimation of Dependences Based on Empirical Data (in Russian)*. Nauka, USSR, 1979.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)*, pages 1063–1069, June 2003a.

R. Vidal, S. Soatto, Y. Ma, and S. Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 1, pages 167–172, December 2003b.

R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1945–1959, December 2005.

H. Wold. Estimation of principal components and related models by iterative least squares. In P.R. Krishnaiah, editor, *Multivariate Analysis*, pages 391–420. New York: Academic Press, 1966.

X. Yang, H. Fu, H. Zha, and J. Barlow. Semi-supervised nonlinear dimensionality reduction. In *Proceedings of the 23rd international conference on Machine learning (ICML '06)*, pages 1065–1072, New York, NY, USA, 2006. ACM.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

J. Zhang, S. Z. Li, and J. Wang. Manifold learning and applications in recognition. *Intelligent Multimedia Processing with Soft Computing*, 2004. Springer-Verlag, Heidelberg.

M. Zibulevsky and M. Elad. L1-L2 optimization in signal and image processing. *Signal Processing Magazine, IEEE*, 27(3):76–88, May 2010.