

Linköping Studies in Science and Technology. Dissertations.
No. 998

Estimation of Nonlinear Dynamic Systems

Theory and Applications

Thomas B. Schön



Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden
Linköping 2006

Estimation of Nonlinear Dynamic Systems – Theory and Applications

© 2006 Thomas B. Schön

*schon@isy.liu.se
www.control.isy.liu.se
Division of Automatic Control
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping
Sweden*

ISBN 91-85497-03-7 ISSN 0345-7524

Printed by LiU-Tryck, Linköping, Sweden 2006

I dedicate this thesis to the memory of my brother Erik

Abstract

This thesis deals with estimation of states and parameters in nonlinear and non-Gaussian dynamic systems. Sequential Monte Carlo methods are mainly used to this end. These methods rely on models of the underlying system, motivating some developments of the model concept. One of the main reasons for the interest in nonlinear estimation is that problems of this kind arise naturally in many important applications. Several applications of nonlinear estimation are studied.

The models most commonly used for estimation are based on stochastic difference equations, referred to as state-space models. This thesis is mainly concerned with models of this kind. However, there will be a brief digression from this, in the treatment of the mathematically more intricate differential-algebraic equations. Here, the purpose is to write these equations in a form suitable for statistical signal processing.

The nonlinear state estimation problem is addressed using sequential Monte Carlo methods, commonly referred to as particle methods. When there is a linear sub-structure inherent in the underlying model, this can be exploited by the powerful combination of the particle filter and the Kalman filter, presented by the marginalized particle filter. This algorithm is also known as the Rao-Blackwellized particle filter and it is thoroughly derived and explained in conjunction with a rather general class of mixed linear/nonlinear state-space models. Models of this type are often used in studying positioning and target tracking applications. This is illustrated using several examples from the automotive and the aircraft industry. Furthermore, the computational complexity of the marginalized particle filter is analyzed.

The parameter estimation problem is addressed for a relatively general class of mixed linear/nonlinear state-space models. The expectation maximization algorithm is used to calculate parameter estimates from batch data. In devising this algorithm, the need to solve a nonlinear smoothing problem arises, which is handled using a particle smoother. The use of the marginalized particle filter for recursive parameter estimation is also investigated.

The applications considered are the camera positioning problem arising from augmented reality and sensor fusion problems originating from automotive active safety systems. The use of vision measurements in the estimation problem is central to both applications. In augmented reality, the estimates of the camera's position and orientation are imperative in the process of overlaying computer generated objects onto the live video stream. The objective in the sensor fusion problems arising in automotive safety systems is to provide information about the host vehicle and its surroundings, such as the position of other vehicles and the road geometry. Information of this kind is crucial for many systems, such as adaptive cruise control, collision avoidance and lane guidance.

Sammanfattning

Denna avhandling behandlar skattning av tillstånd och parameterar i olinjära och icke-gaussiska system. För att åstadkomma detta används huvudsakligen sekventiella Monte Carlo-metoder. Dessa metoder förlitar sig på modeller av det underliggande systemet, vilket motiverar vissa utvidgningar av modellkonceptet. En av de viktigaste anledningarna till intresset för olinjär skattning är att problem av detta slag uppstår naturligt i många viktiga tillämpningar. Flera tillämpade olinjära skattningsproblem studeras.

De modeller som används för skattning är normalt baserade på stokastiska differensekvationer, vanligtvis kallade tillståndsmodeller. Denna avhandling använder huvudsakligen modeller av detta slag. Ett undantag utgörs dock av de matematiskt mer komplikrade differential-algebraiska ekvationerna. Målet är i detta fall att skriva om ekvationerna på en form som lämpar sig för statistisk signalbehandling.

Det olinjära tillståndsskattningsproblemet angrips med hjälp av sekventiella Monte Carlo-metoder, även kallade partikelmetoder. En linjär substruktur ingående i den underliggande modellen kan utnyttjas av den kraftfulla kombinationen av partikelfiltret och kalmanfiltret som tillhandahålls av det marginaliserade partikelfiltret. Denna algoritm går även under namnet Rao-Blackwelliserat partikelfilter och den härleds och förklaras för en generell klass av tillståndsmodeller bestående av såväl linjära, som olinjära ekvationer. Modeller av denna typ används vanligen för att studera positionerings- och målföljnings-tillämpningar. Detta illustreras med flera exempel från fordons- och flygindustrin. Vidare analyseras även beräkningskomplexiteten för det marginaliserade partikelfiltret.

Parameterskattningsproblemet angrips för en relativt generell klass av blandade linjära/olinjära tillståndsmodeller. "Expectation maximization"-algoritmen används för att beräkna parameterskattningar från data. När denna algoritm appliceras uppstår ett olinjärt glättningsproblem, vilket kan lösas med en partikelglättare. Användandet av det marginaliserade partikelfiltret för rekursiv parameterskattning undersöks också.

De tillämpningar som betraktas är ett kamerapositioneringsproblem härstammande från utökad verklighet och sensor fusionproblemet som uppstår i aktiva säkerhetssystem för fordon. En central del i båda dessa tillämpningar är användandet av mätningar från kamerabilder. För utökad verklighet används skattningarna av kamerans position och orientering för att i realtid överlägra datogenererade objekt i filmsekvenser. Syftet med sensor fusionproblemet som uppstår i aktiva säkerhetssystem för bilar är att tillhandahålla information om den egna bilen och dess omgivning, såsom andra fordons positioner och vägens geometri. Information av detta slag är nödvändig för många system, såsom adaptiv farthållning, automatisk kollisionsundvikning och automatisk filföljning.

Acknowledgments

During my work with this thesis I have met and interacted with many interesting people who have, in one way or another, influenced the path of my research. First of all I would like to express my deepest gratitude towards Professor Fredrik Gustafsson, who has been my thesis advisor during the past four years. He is a never-ending source of inspiration and ideas and I just wish I could make use of them all. Furthermore, I very much appreciate his great enthusiasm for the subject and his cheerful attitude. We have had a lot of fun over the years. I truly enjoy working with you, Fredrik!

I am very grateful to Professor Lennart Ljung for creating an excellent environment for conducting research. A special thanks goes to Dr. Jan Maciejowski and Professor Lennart Ljung for introducing me to the world of academic research during my time at the University of Cambridge in Cambridge, United Kingdom in 2001. Without that pleasant stay, this thesis would not exist today.

I would also like to thank Brett Ninness for inviting me to the University of Newcastle in Newcastle, Australia. During my time there I had many interesting experiences, ranging from research discussions on nonlinear estimation to diving with sharks. My office mates, Dr. Adrian Wills and Dr. Sarah Johnson were great and they really made me feel at home. I thank Professor Tomoyuki Higuchi for inviting me to the Institute of Statistical Mathematics in Tokyo, Japan on my way to Australia.

Several colleges deserve special thanks for always taking their time to listen to my ideas and answering my questions. Dr. Fredrik Tjärnström, for being my scientific mentor in the beginning of my studies. We have had and still have many interesting discussions regarding research strategies, system identification and many non-scientific topics as well. Dr. Martin Enqvist, for our interesting discussions. Jeroen Hol, for a good collaboration on inertial sensors and image processing. Ulla Salaneck, for being the wonderful person she is and for always helping me with administrative issues. Gustaf Hendeby, for always helping me in my constant trouble of getting LaTeX to do what I want.

I am very grateful to my co-authors for all the interesting discussions we have had while carrying out the research leading to our publications. They are (in alphabetical order), Andreas Eidehall, Markus Gerdin, Professor Torkel Glad, Professor Fredrik Gustafsson, Dr. Anders Hansson, Dr. Rickard Karlsson, Professor Lennart Ljung, Brett Ninness, Per-Johan Nordlund and Dr. Adrian Wills.

Parts of the thesis have been proofread by Andreas Eidehall, Markus Gerdin, Professor Fredrik Gustafsson, Gustaf Hendeby, Jeroen Hol, Dr. Rickard Karlsson, Martin Ohlson, Henrik Tidefelt and Dr. Fredrik Tjärnström. Your comments and suggestions have improved the quality of the thesis substantially and I am very grateful for that. I am responsible for any remaining errors.

During my work with this thesis I have been involved in two applied research projects, Markerless real-time Tracking for Augmented Reality Image Synthesis (MATRIS) and SEnsor Fusion for Safety systems (SEFS). This has provided me with very valuable insights into the differences and similarities of applied and more theoretical research. I would like to thank the partners; AB Volvo, Volvo Car Corporation, Mecel, Chalmers University of Technology, Linköping University, Fraunhofer IGD, BBC R&D, Christian-Albrechts University and Xsens Technologies B.V. for all the discussions and work leading to these insights.

The financial support for my research has been provided by the Swedish Research Council, the SSF graduate school ECSEL, the European Union, via the MATRIS project and the Intelligent Vehicle Safety Systems (IVSS) program, via the SEFS project. This is truly appreciated.

Dr. Ragnar Wallin has a black belt in staying cool, which he uses in his constant struggle of trying to teach me the art of keeping calm. This has proven useful to me, since I have a slight tendency of getting rather (over-)excited about things. Furthermore, we share the common interest of enjoying good wines and having a beer or two on Thursdays. Martin Ohlson, a good friend, has always taken his time to have an espresso with me, discussing anything from multivariate statistics to squash. I also thank my good friend David Broman for engaging discussions on research and most importantly for all the fun we have had over the years.

Finally, and most importantly, I would like to thank my mother Karin, my father Stefan, my brother Sven and my wonderful friends for their incredible support when the real life has been tough. I owe it all to you.

Linköping, December 2005

Thomas B. Schön

Contents

1	Introduction	1
1.1	Automotive Navigation – Strategy	2
1.2	Automotive Navigation – Example	3
1.2.1	Dynamic Model	4
1.2.2	State Estimation	5
1.3	Navigation for Augmented Reality	7
1.4	Mathematical Background	10
1.5	Outline	11
1.5.1	Outline of Part I	11
1.5.2	Outline of Part II	11
1.6	Contributions	16
I	Topics in Nonlinear Estimation	19
2	Models of Dynamic Systems	21
2.1	Introduction	22
2.2	Preparing for State-Space Models	23
2.3	State-Space Models	25
2.3.1	Nonlinear State-Space Models	25
2.3.2	Mixed Linear/Nonlinear State-Space Models	27
2.3.3	Linear State-Space Models	29
2.4	Linear Differential-Algebraic Equations	29
3	Nonlinear State Estimation	31
3.1	Brief History of the State Estimation Problem	32
3.2	Conceptual Solution	33

3.3	Point Estimates	35
3.4	Nonlinear Systems	36
3.4.1	Local Approximations	37
3.4.2	Global Approximations	39
3.5	Linear Systems	40
3.5.1	Filtering and Prediction	40
3.5.2	Smoothing	42
3.6	Improved Estimation Using Change Detection	43
3.7	Convex Optimization for State Estimation	45
3.7.1	Deterministic Approach to State Estimation	45
3.7.2	Constrained State Estimation	47
4	Sequential Monte Carlo Methods	51
4.1	Perfect Sampling	52
4.2	Random Number Generation	53
4.2.1	Sampling Importance Resampling	53
4.2.2	Acceptance – Rejection Sampling	55
4.2.3	Metropolis – Hastings Independence Sampling	56
4.3	Particle Filter	56
4.3.1	Resampling Algorithms	58
4.3.2	Algorithm Modifications	60
4.3.3	Implementation	61
4.4	Marginalized Particle Filter	63
4.5	Particle Smoother	65
4.5.1	A Particle Smoothing Algorithm	65
4.5.2	Alternative Particle Smoothing Algorithm	67
4.6	Obtaining the Estimates	67
5	Nonlinear System Identification	69
5.1	System Identification Problem	69
5.2	Model Estimation	70
5.2.1	Overview	71
5.2.2	Expectation Maximization Algorithm	72
5.3	Approaches Based on Particle Methods	74
5.3.1	Marginalized Particle Filter	75
5.3.2	Expectation Maximization and the Particle Smoother	75
5.3.3	Discussion	76
6	Concluding Remarks	77
6.1	Conclusion	77
6.2	Future Research	78
A	Appendix, Proof of Corollary 3.1	79
Bibliography		83

II Publications	101
A Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models	103
1 Introduction	106
2 Marginalization	108
2.1 Standard Particle Filter	108
2.2 Diagonal Model	109
2.3 Triangular Model	111
2.4 General Case	113
3 Important Special Cases and Extensions	115
3.1 Generalized Noise Assumptions	116
3.2 Important Model Class	116
4 Illustrating Example	117
5 Integrated Aircraft Navigation	118
5.1 Dynamic Model	119
5.2 Result	120
6 Conclusion	122
A Proof for Theorem A.1	123
References	125
B Complexity Analysis of the Marginalized Particle Filter	129
1 Introduction	131
2 Marginalized Particle Filter	132
3 Complexity Analysis	133
3.1 Nonlinear Measurements	134
3.2 Mixed Nonlinear/Linear Measurements	135
4 Target Tracking Example	136
4.1 Numerical Complexity Analysis	137
4.2 Simulation – Constant Time	139
4.3 Simulation – Constant Velocity RMSE	139
5 Conclusion	140
References	140
C A Modeling and Filtering Framework for Linear Differential-Algebraic Equations	143
1 Introduction	145
2 Derivation of the Process Noise Subspace	147
2.1 Time Domain Derivation	147
2.2 Frequency Domain Derivation	149
3 Filtering	150
3.1 Discretization	150
3.2 Kalman Filter	151
4 Example	151
5 Discrete-Time Linear Descriptor Systems	153
5.1 Frequency Domain	154

6	Conclusion	155
7	Acknowledgment	155
	References	155
D	A Note on State Estimation as a Convex Optimization Problem	157
1	Introduction	159
2	Convex Optimization	159
3	Notation and Background	160
4	Convex Optimization Estimation	160
5	Linear Regression Example	161
6	Convex Optimization Filtering	162
7	HMM Example	164
8	Conclusion	166
	References	166
E	Particle Filters for System Identification of State-Space Models Linear in Either Parameters or States	169
1	Introduction	172
2	Particle Filter	173
2.1	Recursive Bayesian Estimation	173
2.2	Implementation	173
2.3	Algorithm	173
3	Marginalization for Variance Reduction	174
4	Models	175
4.1	State-Space Models Linear in the States	175
4.2	State-Space Models Linear in the Parameters	177
5	Chaos Example	179
6	Conclusion	181
	References	181
F	Maximum Likelihood Nonlinear System Estimation	183
1	Introduction	185
2	Problem Formulation	186
3	Expectation Maximization Algorithm	187
4	EM for Parameter Estimation	188
5	Monte Carlo Based Smoothing	189
5.1	Random Sampling	189
5.2	Monte Carlo Based Filtering	190
5.3	Particle Smoother	191
5.4	Using a Particle Smoother with EM	192
6	Simulation Example	193
7	Conclusion	195
	References	197

G Integrated Navigation of Cameras for Augmented Reality	199
1 Introduction	201
2 Dynamic Motion Model	202
2.1 Geometry and Co-ordinate Systems	203
2.2 Position	203
2.3 Orientation	204
3 Using Vision as a Sensor	206
3.1 Camera Model	206
3.2 Two Dimensional Feature Displacements	206
3.3 Three Dimensional Features and Model	208
4 Illustration	209
5 Conclusion	210
6 Acknowledgment	210
References	210
H The Marginalized Particle Filter in Practice	213
1 Introduction	215
2 Marginalized Particle Filter	217
2.1 Representation	217
2.2 Model	218
2.3 Algorithm	218
3 Analysis	221
3.1 Variance Reduction	221
3.2 Computational Complexity	221
3.3 Quantization Effects	223
4 Introducing the Applications	225
5 Positioning Applications	227
5.1 Underwater Terrain-aided Positioning	227
5.2 Aircraft Terrain-Aided Positioning	228
5.3 Automotive Map-Aided Positioning	230
6 Target Tracking Applications	230
6.1 Automotive Target Tracking	230
6.2 Bearings-Only Target Tracking	231
6.3 Radar Target Tracking	232
7 Concluding Discussion	234
References	235
I Lane Departure Detection for Improved Road Geometry Estimation	239
1 Introduction	241
2 Estimation Problem	244
3 Detecting Lane Departures	245
3.1 Distance Measure	246
3.2 Stopping Rule	247
3.3 Application and Result	247
4 Further enhancement	248
5 Alternative methods	249

6	Conclusion	251
7	Acknowledgment	251
	References	253
	Index	261

1

Introduction

THIS thesis is concerned with the problem of estimating various quantities in nonlinear dynamic systems. The ability to handle this problem is of paramount importance in many practical applications. In order to understand how a system, for instance, a car, an aircraft, a spacecraft or a camera performs, we need to have access to certain important quantities associated with the system. Typically we do not have direct access to these, implying that they have to be estimated based on various noisy measurements available from the system. Both theoretical developments and application oriented studies are presented. The interplay between the theory and application provides interesting and valuable insights and it prevents us from developing fallacies concerning the relative importance of various theoretical concepts, allowing for a balanced view. Furthermore, it enables a systematic treatment of the applications.

This first chapter illustrates the kind of problems that can be handled using the theory developed in this thesis, by explaining two applications. The first applications stems from the automotive industry, where the current development of active safety systems require better use of the available sensor information. The second applications deals with the problem of estimating the position and orientation of a camera, using information from inertial sensors and computer vision. Mathematically speaking, the two applications are rather similar, they both result in nonlinear estimation problems. Another common characteristic is that information from several different sensors have to be merged or fused. Problems of this kind are commonly referred to as *sensor fusion* problems.

A unified approach to handle the sensor fusion problem arising in automotive safety systems is introduced in Section 1.1 and exemplified in Section 1.2. The second application is introduced in Section 1.3. In Section 1.4 we provide a brief mathematical background to the problem under study. The outline is provided in Section 1.5. Finally, the chapter is concluded with a statement of the contributions in Section 1.6.

1.1 Automotive Navigation – Strategy

The automotive industry is an industry in change, where the focus is currently shifting from mechanics to electronics and software. To quantify this statement the monetary value of the software in a car is predicted to increase from 4% in 2003, to 13% in 2010 (Forssell and Gustafsson, 2004). The key reason for this substantial increase is the rather rapid development of automotive safety systems (Gustafsson, 2005). This opens up for many interesting applications and research opportunities within the field of estimation theory.

Automotive safety systems are currently serving as a technological driver in the development and application of estimation theory, very much in the same way that the aerospace industry has done in the past. In fact, the automotive industry is currently faced with several of the problems already treated by the aerospace industry, for example collision avoidance and navigation. Hence, a lot can probably be gained in reusing results from the latter in solving the problems currently under investigation in the former. The development within the aerospace industry is reviewed by McGee and Schmidt (1985). Within the next 10–20 years there will most certainly be similar reviews written, treating the development within the automotive industry, indeed an early example of this is Gustafsson (2005).

The broadest categorization of automotive safety systems is in terms of *passive* and *active* systems. Passive systems are designed to mitigate harmful effects *during* accidents. Examples include seat belts, air bags and belt pretensioners. The aim of active systems is to prevent accidents *before* they occur. To mention some examples of active systems, we have ABS (Anti-lock Braking System), ACC (Adaptive Cruise Control) and collision avoidance. More thorough reviews of existing and future systems are given in Eidehall (2004), Jansson (2005), Danielsson (2005), Gustafsson (2005). There is an interesting study by Eidehall (2004), where different potential active safety systems are profiled with respect to accident statistics, system complexity and cost.

The current situation within the automotive industry is that each control system, read active safety system, comes with the necessary sensors. Each sensor belongs to a certain control system and it is only used by this system. This effectively prevents other systems from using the, potentially very useful, information delivered by the sensor. This situation is most likely to be changed in the future, concurrently with the introduction of more control systems in cars. A unifying feature of all control systems is that they rely on accurate state¹ information. As Gustafsson (2005) points out, it is currently *more important to have accurate state information than advanced control algorithms*. Indeed, it is often sufficient to employ simple P(I)D controllers. Hence, it is more important what information to feed back than how the actual feedback is performed.

The natural conclusion from the discussion above is that the data from the different sensors should be jointly analyzed to produce the best possible estimate of the state. The state information can then be accessed by all control systems in the cars. This idea is briefly illustrated in Figure 1.1. This approach is employed in the applied research

¹Depending on which control system we are concerned with the state is obviously different. In the example given in the subsequent section, the state contains information about the motion of the host vehicle and the surrounding vehicles and the road geometry.

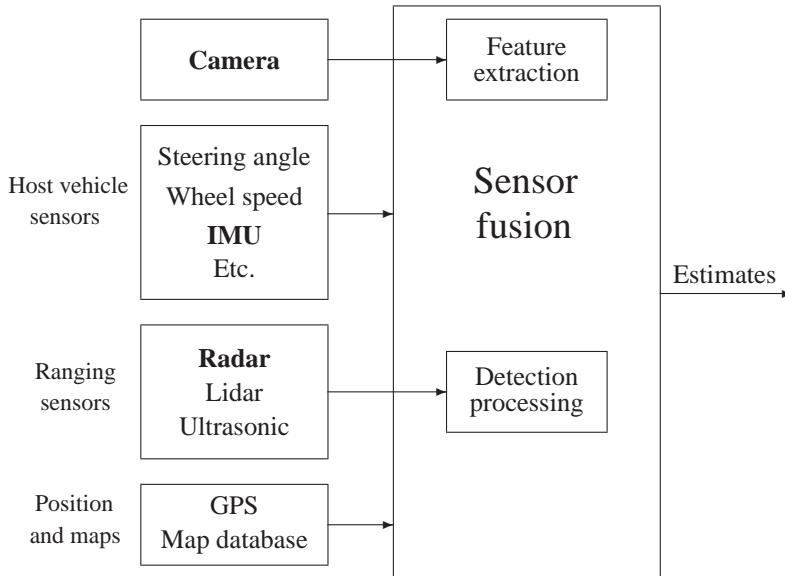


Figure 1.1: The most important factor enabling future automotive safety systems is the availability of accurate information about the state. The process of obtaining this information is to a large extent dependent on a unified treatment of the sensor information, as illustrated in this figure. The aim of this sensor fusion approach is to provide the best information possible for as many purposes as possible. In Section 1.2 this strategy is exemplified using the sensors in bold font.

project, SEFS², where we take part. Similar ideas have previously been suggested, for instance by Streller et al. (2002). The figure does not claim to contain an exhaustive list of possible sensors, it is merely intended as an illustration of the idea. For an introduction to automotive sensors, see, for example, Danielsson (2005), Nwagbosso (1993), Strobel et al. (2005). In the subsequent section an explicit example is provided, where the idea presented above has been employed and evaluated using authentic traffic data.

1.2 Automotive Navigation – Example

The objective of this study is to calculate estimates of the road geometry, which are important in several advanced control systems such as lane guidance and collision avoidance. The sensors used to accomplish this are primarily radar and camera, with appropriate image processing provided by the supplier. Hence, the idea exemplified here follows from the general framework introduced in Figure 1.1. The result, using authentic traffic data, will illustrate the power of a model based sensor fusion approach. Here, information

²SEnsor Fusion for Safety systems (SEFS) is an applied research project, with participants from AB Volvo, Volvo Car Corporation, Mecel, Chalmers University of Technology and Linköping University. The financial support is provided by the Intelligent Vehicle Safety Systems (IVSS) program.

from several sensors is used to obtain better performance, than separate use of the sensors would allow for. The vision system delivers estimates of the road geometry, but the quality of these estimates is not sufficient for future automotive safety systems. The idea is to improve the quality by using information available from the motion of the surrounding vehicles, measured using the radar, together with information from the vision system. The key assumption is that the leading vehicles will keep following their lane, and their lateral movement can thus be used to support the otherwise difficult process of road geometry estimation. For example, when entering a curve as in Figure 1.2 the vehicles ahead will start moving to the right and thus there is a high probability that the road is turning to



Figure 1.2: When entering a curve, all vehicles start moving in the lateral direction. This information can be used to support the road geometry estimate.

the right. This information, obtained from radar measurements, can be used to significantly improve the rather crude road geometry estimates from the vision system. This idea of jointly estimating the position of the surrounding vehicles and the road parameters has previously been successfully applied, see, e.g., Eidehall (2004), Dellaert and Thorpe (1997), Zomotor and Franke (1997), but as will be explained in the sequel the estimates can be further enhanced.

In the subsequent sections this problem will be posed as an estimation problem, which can be solved using the model based estimation algorithms presented in this thesis. First of all a dynamic model is derived. More specifically, the resulting model is a mixed linear/nonlinear state-space model, to be described in Chapter 2. The state estimation problem arising from models in this form can be handled using either the marginalized particle filter, thoroughly derived in Paper A, or the extended Kalman filter (EKF).

1.2.1 Dynamic Model

Dynamic motion models for various objects have been extensively studied and the literature contains hundreds of papers describing different models, bearing names like constant velocity model, constant acceleration model, coordinated turn model, etc. The resulting

models are all expressed in the general classes introduced in Chapter 2. There are several surveys available, dealing with various motion models, see, e.g., Bar-Shalom and Li (1993), Li and Jilkov (2003, 2001), Blackman and Popoli (1999).

For the present study we need models describing the motion of the host vehicle, the surrounding vehicles and the road. In the host vehicle we have access to sensors measuring wheel speed, yaw rate, steering wheel angle, etc. This allows for a more detailed model of the host vehicle, than what can be devised for the surrounding vehicles. We will make use of the model derived by Eidehall (2004). For the present discussion it is only the lateral motion model of the surrounding vehicles which is important. Further details concerning the model are given in the Appendix of Paper I. The essential feature of the model is that it is based on a curved coordinate system, which is attached to the road. This will enable the use of very simple models for the surrounding vehicles. The key assumption introduced above, that the surrounding vehicles will keep following the same lane, is in discrete-time expressed as $y_{t+1}^i = y_t^i + w_t, w_t \sim \mathcal{N}(0, Q_{\text{lat}})$. Here, y^i denotes the lateral position of vehicle i and w_t denotes Gaussian white noise which is used to account for model uncertainties.

1.2.2 State Estimation

The resulting nonlinear state estimation problem can be solved using either the extended Kalman filter (Eidehall and Gustafsson, 2004) or the marginalized particle filter (Eidehall et al., 2005). For the present study the extended Kalman filter has been employed. The estimate of the road curvature during an exit phase of a curve is illustrated in Figure 1.3. To facilitate comparison, the true reference signal and the raw vision measurement of the

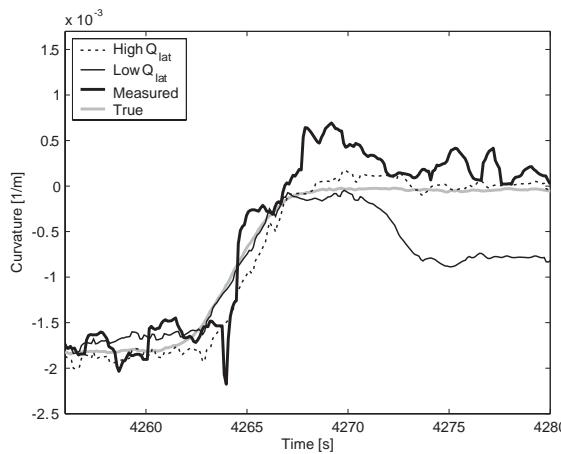


Figure 1.3: Comparison of estimation performance from two filters, one with a large Q_{lat} and one with a small Q_{lat} . The raw measurement signal from the image processing unit is also included. Comparing this raw vision measurement to the result from the filters clearly illustrates the power of a model based sensor fusion approach.

curvature are included as well. The true reference signal was generated using the method proposed by Eidehall and Gustafsson (2006). Comparing this raw vision measurement to the result from the filters clearly illustrates the power of a model based sensor fusion approach. In this particular scenario there are two leading vehicles used to support the curvature estimates, see Figure 1.2.

From Figure 1.3 it is clear that the filter with a low value of Q_{lat} performs much better, than the filter with a high value of Q_{lat} , during the curve exit. This suggests that the filter should be tuned using a low value for Q_{lat} . However, at time 4270 s, when the road is straight, the performance of this filter deteriorates. If the recorded video is studied, see Figure 1.4, it can be seen that this performance degradation coincides exactly with a



Figure 1.4: A snapshot from the video just after time 4270 s, when the lane change of the tracked vehicle commences.

lane change of one of the leading vehicles. Obviously, this lane change violates the key assumption, that the leading vehicles will keep driving in the same lane. In fact, all lateral movements, such as lane changes, performed by the leading vehicle will be interpreted as a turn in the road by the present approach. However, the filter using a larger value of Q_{lat} does not suffer from this problem. This is natural, since a higher value of Q_{lat} corresponds to that the model allows for larger lateral movements of the leading vehicles. On the other hand, since this model contains more noise than necessary, the quality of the estimates is bad due to this. This is manifested by the time delay in the estimate during the curve exit and its overall shaky behavior. This is actually an example of the fundamental limitation present in all linear filters; the estimation performance is a compromise between noise attenuation and tracking ability.

Based on the discussion above it is advisable to use a low value for Q_{lat} when the key assumption holds and a larger value for Q_{lat} when it does not hold. This can be achieved by detecting vehicles which violate the key assumption, i.e., performs lane departures, and adapt the model accordingly. This is further investigated in Paper I, where it is shown to result in significantly improved road geometry estimates.

1.3 Navigation for Augmented Reality

The following navigation application stems from the area of augmented reality (AR), where the idea is to overlay virtual, computer generated objects onto an authentic scene in real time. This can be accomplished either by displaying them in a see-through head-mounted display or by superimposing them on the images from a camera. There are many applications for augmented reality, ranging from broadcasting and film production, to industrial maintenance, medicine, entertainment and games, see Figure 1.5 for some examples. For a survey of the field, see, e.g., Azuma (1997), Azuma et al. (2001).



Figure 1.5: Some examples illustrating the concept of augmented reality.

One of the key enabling technologies for augmented reality is to be able to determine the position and orientation of the camera, with high accuracy and low latency. To accomplish this there are several sensors which can be used, see Welch and Foxlin (2002) for an overview. Accurate information about the position and orientation of the camera is essential in the process of combining the real and the virtual objects. Prior work in this recent research area have mainly considered the problem in an environment which has been prepared in advance with various artificial markers, see, e.g., Thomas et al. (1997), Caarls et al. (2003), Yokokohji et al. (2000), You and Neumann (2001). The current trend is to shift from prepared to unprepared environments, which makes the problem much harder. On the other hand, the costly procedure of preparing the environment with markers will no

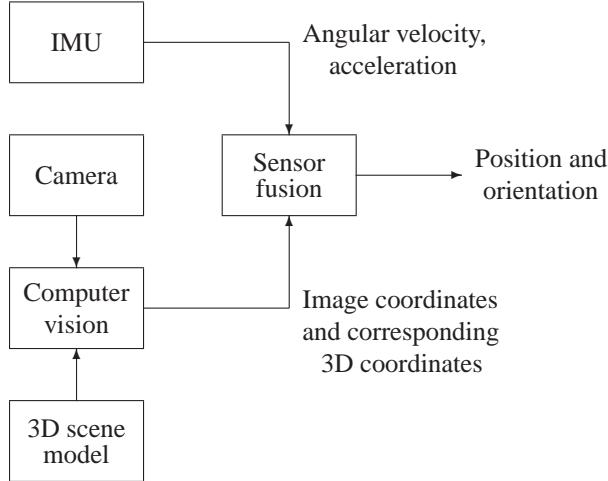


Figure 1.6: Schematic illustration of the approach. The sensor fusion module is basically a recursive nonlinear state estimator, using information from the inertial measurement unit (IMU) and the computer vision system to compute an estimate of the position and orientation of the camera.

longer be required. Furthermore, in outdoor situations it is generally not even possible to prepare the environment with markers. The idea is to make use of natural features, occurring in the real scene, as markers. This problem of estimating the camera's position and orientation in an unprepared environment has previously been discussed in the literature, see, e.g., Simon and Berger (2002), Lepetit et al. (2003), Genc et al. (2002), You et al. (1999), Klein and Drummond (2003). Furthermore, the work by Davison (2003), Davison et al. (2004) is interesting in this context. Despite all the current research within the area, the objective of estimating the position and orientation of a camera in an unprepared environment still presents a challenging problem.

The problem introduced above can in fact be cast as a nonlinear state estimation problem. This work is performed within a consortium, called MATRIS (2005)³, where the objective is to solve this estimation problem in an unprepared environment, using the information available in the camera images and the accelerations and angular velocities delivered by an inertial measurement unit (IMU). A schematic illustration of the approach is given in Figure 1.6. The IMU, which is attached to the camera, provides measurements of the acceleration and the angular velocity of the camera. The accelerometers and the gyroscopes used to obtain these measurements are of MEMS type, implying small, low cost sensors. However, these sensors are only reliable on a short time scale, due to an inherent drift. This drift is compensated for using information from the computer vision system,

³Markerless real-time Tracking for Augmented Reality Image Synthesis (MATRIS) is the name of a sixth framework research program, funded by the European Union (EU), contract number: IST-002013. It is an interdisciplinary applied research project with the following partners; Fraunhofer IGD, BBC R&D, Christian-Albrechts University, Xsens Technologies B.V. and Linköping University.

which consists of a 3D scene model and real time feature extraction. The 3D model is generated off-line using images of the scene or existing CAD models (Koch et al., 2005). It contains positions of various natural markers, which are then detected in the images using feature extraction techniques. This allows the computer vision system to deliver the 3D coordinates of a natural marker, together with the corresponding coordinates for this marker in the present image. This information is then used together with the information from the IMU in order to compute an estimate of the position and orientation of the camera. This computation is performed in the sensor fusion block in Figure 1.6. Hence, sensor fusion is interpreted as the process of forming an appropriate nonlinear state estimation problem, which can be solved in real time, using the available sensor information as efficient as possible. For further details regarding this approach, see Paper G and Hol (2005).

The simultaneous use of information present in images and information from inertial sensors is currently under investigation within many branches of science and there exists a vast amount of interesting application areas. In the previous section it was illustrated that this is a sub-problem arising in the development of automotive safety systems. A useful prototype for investigating this problem has been developed in the MATRIS project, see Figure 1.7. By using the data from this prototype together with the simultaneous lo-



Figure 1.7: This is a prototype developed in the MATRIS project. It consists of a camera, an IMU and a low-power digital signal processor, used for pre-processing of the sensor signals. Courtesy of Xsens Technologies B.V.

calization and mapping (SLAM) ideas of Davison (2003) it should be possible to derive rather good estimates. Furthermore, the presence of the inertial information will probably allow for the use of simple image processing. Perhaps very simple point-of-interest (POI) detectors such as the Harris detector, introduced by Harris and Stephens (1988), can be used. Another interesting observation elaborated upon by Huster (2003) is that the vision measurements can be interpreted as bearing measurements. This opens up for reuse of the research performed on the bearings-only problem, see, e.g., Karlsson and Gustafsson (2005) for an introduction to this problem using radar, sonar and infrared measurements.

1.4 Mathematical Background

In the previous sections two applications were introduced, both resulting in a *sensor fusion* problem, where the objective is to utilize existing and affordable sensors to extract as much information as possible. The framework for nonlinear state estimation discussed in this thesis provides a systematic approach to handle sensor fusion problems. This thesis will, to a large extent, make use of a probabilistic framework in dealing with estimation problems of this kind. The expressive power of probability density functions opens up for a rather systematic treatment of the estimation problem, where the main ideas can be conveyed, without getting lost in tedious matrix calculations. More specifically, we will make extensive use of the theory originating from the work of the English Reverend Thomas Bayes, published two years after his death in Bayes (1763). The distinguishing feature of the Bayesian theory is that all unknown variables are considered to be random variables. In the classical theory, represented by Fisher (1912, 1922) and his method of *maximum likelihood* the parameters to be estimated are treated as unknown constants. In the literature there is a lively debate, concerning the two viewpoints, represented by Bayes and Fisher, which has been going on for almost a century now. Some good entry points into this debate are provided by Box and Tiao (1992), Edwards (1992), Spall (1988), Robert (2001). We will adopt a rather pragmatic viewpoint, implying that the focus is on using the best approach for each problem, without getting too involved in the philosophical discussions inherent in the debate mentioned above. The Bayesian theory is extensively used in discussing the state estimation theory. On the other hand, Fisher's method of maximum likelihood is employed in solving certain system identification problems. The probabilistic framework for solving estimation problems is indeed very powerful. However, despite this, it is still fruitful to consider the estimation problem as a deterministic problem of minimizing errors. In fact, the two approaches are not as far apart as one might first think.

The estimation problems are handled using *model based* methods. The systems under study are dynamic, implying that the models will mostly be of dynamic nature as well. More specifically, the models are primarily constituted by stochastic difference equations. The most commonly used model is the nonlinear state-space model and various special cases thereof. The nonlinear state-space model consists of a system of nonlinear difference equations according to

$$x_{t+1} = f(x_t, u_t, \theta) + w_t, \quad (\text{System model}) \quad (1.1a)$$

$$y_t = h(x_t, u_t, \theta) + e_t, \quad (\text{Measurement model}) \quad (1.1b)$$

where x_t denotes the state variable, u_t denotes the known input signal, θ denotes the static parameters, y_t denotes the measurements, w_t and e_t denote the process and measurement noise, respectively. The *system model* (1.1a) describes the evolution of the state variables over time, whereas the *measurement model* (1.1b) explains how the measurements relate to the state variables. The dynamic model must describe the essential properties of the underlying system, but it must also be simple enough to make sure that it can be used to devise an efficient estimation algorithm. In tackling the nonlinear state estimation problem it is imperative to have a good model of the system at hand, probably more important than in the linear case. If the model does not provide an adequate description of the underlying system, it is impossible to derive an appropriate estimation algorithm.

It is, surprisingly enough, possible to derive expressions for the complete solution to the nonlinear state estimation problem. However, there is a severe limitation inherent in these expressions, they involve multidimensional integrals which only permit closed-form solutions in certain special cases. The most important special case occurs when all equations are linear and the noise terms are Gaussian in (1.1). The solution is in this case provided by the *Kalman filter* introduced by Kalman (1960). In the nonlinear, non-Gaussian case approximate techniques have to be employed. A common idea is to approximate the nonlinear model by a linear model and then use the Kalman filter for this linearized model, resulting in the extended Kalman filter. There are many applications where this renders acceptable performance, but there are also cases where the resulting state estimates diverge. Furthermore, conceptually it is not a satisfactory solution, since in a way it is solving the wrong problem. A solution, which is conceptually more appealing can be obtained by keeping the nonlinear model and trying to approximate the optimal solution. The reason is that the effort is now spent on trying to solve the correct problem. There is a class of methods, referred to as *sequential Monte Carlo methods*, available for doing this. A popular member of this class is the particle filter, introduced by Gordon et al. (1993). An attractive feature with these methods is, as was noted above, that they provide *an approximate solution to the correct problem, rather than an optimal solution to the wrong problem*. The sequential Monte Carlo methods constitute an important part of this thesis. They will be employed both for the nonlinear state estimation problem and the nonlinear system identification problem.

1.5 Outline

There are two parts in this thesis. The objective of the first part is to give a unified view of the research reported in this thesis. This is accomplished by explaining how the different publications in Part II relate to each other and to the existing theory.

1.5.1 Outline of Part I

This thesis is concerned with estimation methods that employ dynamic models of the underlying system in order to calculate the estimates. In order to be able to use these methods there is of course a need for appropriate mathematical models. This motivates the discussion on various model classes in Chapter 2. A rather general account of the state estimation theory is given in Chapter 3. The sequential Monte Carlo methods are then reviewed in Chapter 4. The nonlinear system identification problem is treated in Chapter 5, where special attention is devoted to the use of the expectation maximization algorithm. Finally, Chapter 6 provide concluding remarks consisting of conclusions and some ideas for future research.

1.5.2 Outline of Part II

This part consists of a collection of edited papers, introduced below. Besides a short summary of the paper, a paragraph briefly explaining the background and the contribution is provided. The background is concerned with how the research came about, whereas the

contribution part states the contribution of the present author. In Table 1.1 the papers are grouped according to the nature of their main content.

Table 1.1: Grouping of the papers according to the nature of their main content.

Content	Paper
Theory, state estimation	A, B, C, D
Theory, system identification	E, F
Applications	G, H, I

Paper A: Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models

Schön, T., Gustafsson, F., and Nordlund, P.-J. (2005). Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289.

Summary: The particle filter offers a general numerical tool to approximate the filtering density function for the state in nonlinear and non-Gaussian filtering problems. While the particle filter is fairly easy to implement and tune, its main drawback is that it is quite computer intensive, with the computational complexity increasing quickly with the state dimension. One remedy to this problem is to marginalize out the states appearing linearly in the dynamics. The result is that one Kalman filter is associated with each particle. The main contribution in this paper is to derive the details for the marginalized particle filter for a general nonlinear state-space model. Several important special cases occurring in typical signal processing applications are also discussed. The marginalized particle filter is applied to an integrated navigation system for aircraft. It is demonstrated that the complete high-dimensional system can be based on a particle filter using marginalization for all but three states. Excellent performance on real flight data is reported.

Background and contribution: The results from Nordlund (2002) have been extended and improved. The author of this thesis wrote the major part of this paper. The example, where the theory is applied using authentic flight data, is the result of the Master’s thesis by Frykman (2003), which the authors jointly supervised.

Paper B: Complexity Analysis of the Marginalized Particle Filter

Karlsson, R., Schön, T., and Gustafsson, F. (2005). Complexity analysis of the marginalized particle filter. *IEEE Transactions on Signal Processing*, 53(11):4408–4411.

Summary: In this paper the computational complexity of the marginalized particle filter, introduced in Paper A, is analyzed and a general method to perform this analysis is given. The key is the introduction of the equivalent flop measure. In an extensive Monte Carlo

simulation different computational aspects are studied and compared with the derived theoretical results.

Background and contribution: Several applications of the marginalized particle filter are discussed in Paper H. During this work the need for a thorough theoretical investigation of the computational complexity of the algorithm was identified, motivating the work reported in this paper. This investigation was carried out in close co-operation with Dr. Rickard Karlsson.

Paper C: A Modeling and Filtering Framework for Linear Differential-Algebraic Equations

Schön, T., Gerdin, M., Glad, T., and Gustafsson, F. (2003a). A modeling and filtering framework for linear differential-algebraic equations. In *Proceedings of the 42nd Conference on Decision and Control*, Maui, Hawaii, USA.

Summary: General approaches to modeling, for instance using object-oriented software, lead to differential-algebraic equations (DAE). For state estimation using observed system inputs and outputs in a stochastic framework similar to Kalman filtering, we need to augment the DAE with stochastic disturbances, “process noise”, whose covariance matrix becomes the tuning parameter. In this paper we determine the subspace of possible causal disturbances based on the linear DAE model. This subspace determines all degrees of freedom in the filter design, and a Kalman filter algorithm is given.

Background and contribution: This paper is the result of work conducted in close co-operation with Markus Gerdin. It provided a start for introducing stochastic processes in differential-algebraic equations. The results have recently been refined by Gerdin et al. (2005a). Finally, a paper presenting the resulting framework for system identification and state estimation in linear differential-algebraic equations has been submitted to *Automatica* (Gerdin et al., 2005b).

Paper D: A Note on State Estimation as a Convex Optimization Problem

Schön, T., Gustafsson, F., and Hansson, A. (2003b). A note on state estimation as a convex optimization problem. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 61–64, Hong Kong.

Summary: We investigate the formulation of the state estimation problem as a convex optimization problem. The Kalman filter computes the maximum a posteriori (MAP) estimate of the state for linear state-space models with Gaussian noise. We interpret the Kalman filter as the solution to a convex optimization problem, and show that the MAP state estimator can be generalized to any noise with log-concave density function and any combination of linear equality and convex inequality constraints on the state.

Background: This work started as a project in a graduate course in convex optimization held by Dr. Anders Hansson. My thesis advisor Professor Fredrik Gustafsson came up with the idea when he served as opponent for the thesis by Andersson (2002).

Paper E: Particle Filters for System Identification of State-Space Models Linear in Either Parameters or States

Schön, T. and Gustafsson, F. (2003). Particle filters for system identification of state-space models linear in either parameters or states. In *Proceedings of the 13th IFAC Symposium on System Identification*, pages 1287–1292, Rotterdam, The Netherlands. Invited paper.

Summary: The potential use of the marginalized particle filter for nonlinear system identification is investigated. Algorithms for systems which are linear in either the parameters or the states are derived. In these cases, marginalization applies to the linear part, which firstly significantly widens the scope of the particle filter to more complex systems, and secondly decreases the variance in the linear parameters/states for fixed filter complexity. This second property is illustrated in an example of a chaotic model. The particular case of freely parameterized linear state-space models, common in subspace identification approaches, is bilinear in states and parameters, and thus both cases above are satisfied.

Background and contribution: At the ERNSI (European Research Network System Identification) workshop held in Le Croisic, France in 2002 someone mentioned that it would be interesting to investigate if the particle filter can be useful for the system identification problem. This comment, together with the invited session on particle filters held at the 13th IFAC Symposium on System Identification, in Rotterdam, the Netherlands, served as catalysts for the work presented in this paper.

Paper F: Maximum Likelihood Nonlinear System Estimation

Schön, T. B., Wills, A., and Ninness, B. (2006b). Maximum likelihood nonlinear system estimation. In *Proceedings of the 14th IFAC Symposium on System Identification*, Newcastle, Australia. Accepted for publication.

Summary: This paper is concerned with the parameter estimation of a relatively general class of nonlinear dynamic systems. A Maximum Likelihood (ML) framework is employed in the interests of statistical efficiency, and it is illustrated how an Expectation Maximization (EM) algorithm may be used to compute these ML estimates. An essential ingredient is the employment of particle smoothing methods to compute required conditional expectations via a sequential Monte Carlo approach. A simulation example demonstrates the efficacy of these techniques.

Background and contribution: This work is a result of the author's visit to the University of Newcastle in Newcastle, Australia during the period February – May, 2005. It was conducted in close co-operation with Dr. Adrian Wills and Dr. Brett Ninness, both having extensive experience in using the EM algorithm for system identification, whereas the author of this thesis has been working with sequential Monte Carlo methods. We agreed on that it would be interesting to try and combine those ideas in order to tackle a certain class of nonlinear system identification problems.

Paper G: Integrated Navigation of Cameras for Augmented Reality

Schön, T. B. and Gustafsson, F. (2005). Integrated navigation of cameras for augmented reality. In *Proceedings of the 16th IFAC world Congress*, Prague, Czech Republic.

Summary: In augmented reality, the position and orientation of a camera must be estimated very accurately. This paper proposes a filtering approach, similar to integrated navigation in aircraft, which is based on inertial measurements as primary sensor on which dead-reckoning can be based. Features extracted from the image are used as supporting information to stabilize the dead-reckoning. The image features are considered to be sensor signals in a Kalman filter framework.

Background and contribution: This paper is a result of the MATRIS (2005) project, which is an applied interdisciplinary research project. The contents is influenced by the many interesting discussion held during the project meetings around Europe.

Paper H: The Marginalized Particle Filter in Practice

Schön, T. B., Karlsson, R., and Gustafsson, F. (2006a). The marginalized particle filter in practice. In *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, USA. Invited paper, accepted for publication.

Summary: This paper is a suitable primer on the marginalized particle filter, which is a powerful combination of the particle filter and the Kalman filter. It can be used when the underlying model contains a linear sub-structure, subject to Gaussian noise. This paper will illustrate several positioning and target tracking applications, solved using the marginalized particle filter.

Background and contribution: In this paper we have tried to provide a unified inventory of applications solved using the marginalized particle filter. The author of this thesis has been involved in the theoretical background, the computational complexity part and the applications concerned with aircraft terrain-aided positioning, automotive target tracking and radar target tracking.

Paper I: Lane Departure Detection for Improved Road Geometry Estimation

Schön, T. B., Eidehall, A., and Gustafsson, F. (2005). Lane departure detection for improved road geometry estimation. Technical Report LiTH-ISY-R-2714, Department of Electrical Engineering, Linköping University, Sweden. *Submitted to the IEEE Intelligent Vehicle Symposium*, Tokyo, Japan.

Summary: An essential part of future collision avoidance systems is to be able to predict road curvature. This can be based on vision data, but the lateral movement of leading vehicles can also be used to support road geometry estimation. This paper presents a method for detecting lane departures, including lane changes, of leading vehicles. This information is used to adapt the dynamic models used in the estimation algorithm in order

to accommodate for the fact that a lane departure is in progress. The goal is to improve the accuracy of the road geometry estimates, which is affected by the motion of leading vehicles. The significantly improved performance is demonstrated using sensor data from authentic traffic environments.

Background and contribution: The idea for this paper was conceived during one of the authors frequent visits to Göteborg. The work was performed in close co-operation with Andreas Eidehall.

Publication of related interest, but not included in this thesis:

Gerding, M., Schön, T. B., Glad, T., Gustafsson, F., and Ljung, L. (2005b).
On parameter and state estimation for linear differential-algebraic equations.
Submitted to Automatica,

Eidehall, A., Schön, T. B., and Gustafsson, F. (2005). The marginalized particle filter for automotive tracking applications. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 369–374, Las Vegas, USA,

Schön, T. (2003). *On Computational Methods for Nonlinear Estimation*. Licentiate Thesis No 1047, Department of Electrical Engineering, Linköping University, Sweden.

1.6 Contributions

The main contributions are briefly presented below. Since the title of this thesis is *Estimation of Nonlinear Dynamic Systems – Theory and Applications* the contributions are naturally grouped after theory and applications.

Theory

- The derivation of the marginalized particle filter for a rather general mixed linear/nonlinear state-space model. This is presented in Paper A together with a thorough explanation of the algorithm.
- The analysis of the computational complexity of the marginalized particle filter, presented in Paper B.
- A new approach to incorporate white noise in linear differential-algebraic equations is presented in Paper C. This provided the start for a framework allowing for state estimation and system identification in this type of models.
- Two algorithms are introduced to handle the system identification problem occurring in a class of nonlinear state-space models, with affine parameter dependence. In Paper E the marginalized particle filter is employed and in Paper F an algorithm based on a combination of the expectation maximization algorithm and a particle smoothing algorithm is derived.

Applications

- The idea of using feature displacements to obtain information from vision measurements is introduced in Paper G.
- Several applications of the marginalized particle filter are discussed in Paper H.
- A new approach to estimate road geometry, based on change detection, is presented in Paper I.

Part I

Topics in Nonlinear Estimation

2

Models of Dynamic Systems

THE estimation theory discussed in this thesis is model based. Hence, the need for an appropriate model is imperative. By appropriate we mean a model that is well suited for its intended purpose. In other words, when a model is developed it must always be kept in mind what it should be used for. The model must describe the essential properties of the underlying system, but it should also be simple enough to make sure that it can be used to devise an efficient estimation algorithm. If the underlying model is not appropriate it does not matter how good the estimation algorithm is. Hence, a reliable model is essential to obtain good estimates. When we refer to a model, we mean a system of equations describing the evolution of the states and the measurements associated with the application. Other models are for instance impulse responses, transfer functions and Volterra series.

The purpose of this chapter is to provide a hierarchical classification of the most common model classes used here, starting with a rather general formulation. In deriving models for a specific application the need for solid background knowledge of the application should not be underestimated. Several examples of application driven models are given in the papers in Part II. These models are all instances of the general model classes described in this chapter.

The most general model class considered is the *stochastic differential-algebraic equations* (SDAE), briefly introduced in Section 2.1. However, most of the models currently used within the signal processing and automatic control communities are state-space models, which form an important special case of the SDAE model. In Section 2.2 we prepare for the state-space model, which is introduced in Section 2.3. Finally, Section 2.4 concludes the chapter with a discussion on how to include white noise into linear differential-algebraic equations.

2.1 Introduction

The current demand for modularity and more complex models have favored the approach based on *object-oriented modeling*, where the model is obtained by connecting simple sub-models, typically available from model libraries. Examples of modeling tools of this kind are Modelica, Dymola and Omola (Fritzson, 2004, Tiller, 2001, Mattsson et al., 1998). The modeling software will then collect all the equations involved and construct a resulting model, which involves both differential and algebraic equations. A general formulation of such a model is given by

$$F(\dot{z}(t), z(t), \tilde{u}(t), \theta, t) = 0, \quad (2.1)$$

where the dot denotes differentiation w.r.t. time, z denotes the internal variable vector, \tilde{u} denotes the external signals, θ denotes a time-invariant parameter vector and t denotes time. Finally, the dynamics are described by the possibly nonlinear function F , which is a *differential-algebraic equation* (DAE)¹. This introductory discussion is held using continuous-time models, since that is typically where we have to start, due to the fact that most physical phenomena are continuous. However, discrete-time models can be derived from the continuous-time models. In (2.1) there are two important types of external signals \tilde{u} , which have to be treated separately. The first type is constituted by *known* input signals, denoted by u . Typical examples include control signals or measured disturbances. The second type is *unmeasured* inputs, denoted by w . These signals are typically used to model unknown disturbances, which are described using stochastic processes.

A DAE that contains external variables described by stochastic processes will be referred to as a stochastic differential-algebraic equation. There will always be elements of uncertainty in the models, implying that we have to be able to handle SDAEs. As of today there is no general theory available on how to do this. However, several special cases have been extensively studied. In Brenan et al. (1996) and Ascher and Petzold (1998) there is a thorough discussion on deterministic differential-algebraic equations. There has also been some work on stochastic differential-algebraic equations (see, e.g., Winkler, 2003, Schein and Denk, 1998, Penski, 2000, Römisch and Winkler, 2003), but there is still a lot that remains to be done within this field. An intrinsic property of the differential-algebraic equation is that it may hide implicit differentiations of the external signals \tilde{u} . This poses a serious problem if \tilde{u} is described by white noise, because the derivative of white noise is not a well-defined mathematical object. It is thus far from obvious how stochastic processes should be included in this type of equation. In Section 2.4 and Paper C a proposition is given for how to properly incorporate white noise in linear stochastic differential-algebraic equations.

Besides the model for how the system behaves, there is also a need for a model describing how the noisy measurements are related to the internal variables, i.e., a measurement model. Since we cannot measure infinitely often, the measurements are obtained at discrete time instances according to (in the sequel it is assumed that the sampling time is 1 for notational convenience)

$$H(y(t_k), z(t_k), u(t_k), e(t_k), \theta, t_k) = 0, \quad (2.2)$$

¹Other common names for the model class described by (2.1) are implicit systems, descriptor systems, semi-state systems, singular systems, generalized systems, and differential equations on a manifold (Campbell, 1990).

where $y \in \mathbf{R}^{n_y}$ denotes the measurement, $e \in \mathbf{R}^{n_e}$ denotes the measurement noise, t_k denotes the discrete time index, and H denotes a possibly nonlinear function describing how the measurements are obtained. The measurement equation stated in (2.2) is implicit, as opposed to the more specific explicit measurement equation

$$y(t_k) = h(z(t_k), u(t_k), e(t_k), \theta, t_k), \quad (2.3)$$

which is the most common type. However, there are applications implying implicit measurement equations. Examples of this involve positioning systems relying on map information, see, e.g., Gustafsson et al. (2002), Bergman (1999), Hall (2000), Svenzén (2002). Furthermore, measurement equations derived from information in images are sometimes in the form (2.2), which is exemplified in Paper G. By collecting (2.1) and (2.2) a rather general model class can be formulated, the stochastic differential-algebraic equation model.

Model 1 (Stochastic Differential-Algebraic Equation (SDAE) model)

The nonlinear stochastic differential-algebraic equation model is given by

$$F(\dot{z}(t), z(t), u(t), w(t), \theta, t) = 0, \quad (2.4a)$$

$$H(y(t_k), z(t_k), u(t), e(t_k), \theta, t_k) = 0, \quad (2.4b)$$

where $w(t)$ and $e(t_k)$ are stochastic processes.

For a mathematically stricter definition the theory of stochastic differential equations and Itô calculus can be used (Jazwinski, 1970, Øksendal, 2000). However, the definition used here will serve our purposes. As mentioned above the theory on how to handle this quite general stochastic DAE model is far from mature. Several special cases of Model 1 have been extensively studied. The rest of this chapter is devoted to describing some of the most important discrete-time special cases. In fact, most of the models used in the signal processing and the automatic control communities can be considered to be special cases of the rather general formulation in terms of differential-algebraic equations given above. There are of course many different ways to carry out such a classification. We have chosen a classification that we believe serves our purpose best.

An important special case of Model 1 arises when $\dot{z}(t)$ can be explicitly solved for,

$$\dot{z}(t) = f(z(t), u(t), w(t), \theta, t). \quad (2.5)$$

The resulting model is then governed by *ordinary differential equations* (ODE), rather than by differential-algebraic equations. This model is commonly referred to as the continuous-time *state-space model*. To conform with the existing literature the internal variable is referred to as the *state variable* in this special case. Several nonlinear model classes are reviewed by Pearson (1999).

2.2 Preparing for State-Space Models

The discussion in this section is heavily inspired by probability theory. The objective is to provide a transition from the rather general SDAE models discussed in the previous

section to the state-space models introduced in the subsequent section. Note that only discrete-time models are considered and that the possible existence of known input signals u_t is suppressed for brevity.

The *system model* is the dynamic model describing the evolution of the state variables over time. A fundamental property ascribed to the system model is the Markov property.

Definition 2.1 (Markov property). A discrete-time stochastic process $\{x_t\}$ is said to possess the Markov property if

$$p(x_{t+1}|x_1, \dots, x_t) = p(x_{t+1}|x_t). \quad (2.6)$$

In words this means that the realization of the process at time t contains all information about the past, which is necessary in order to calculate the future behavior of the process. Hence, if the present realization of the process is known, the future is independent of the past. This property is sometimes referred to as the *generalized causality principle*, the future can be predicted from knowledge of the present (Jazwinski, 1970). The system model can thus be described as

$$x_{t+1} \sim p_\theta(x_{t+1}|x_1, \dots, x_t) = p_\theta(x_{t+1}|x_t), \quad (2.7)$$

where we have made use of the Markov property. The notation $p_\theta(x)$ is used to describe a family of probability density functions, parameterized by θ . The probability density function $p_\theta(x_{t+1}|x_t)$ describes the evolution of the state variable over time. In general it can be non-Gaussian and include nonlinearities. The initial state is assumed to belong to a probability density function $p_\theta(x_0)$, commonly referred to as the *prior*. Furthermore, the system model can be parameterized by the static parameter θ , as indicated in (2.7). If the parameters are unknown, they have to be estimated before the model can be used for its intended purpose. The task of finding these parameters based on the available measurements is known as the *system identification* problem, which is introduced in Chapter 5. Furthermore, various aspects of the system identification problem are discussed in Paper E and Paper F.

The state process $\{x_t\}$ is an unobserved (hidden) Markov process. Information about this process is indirectly obtained from measurements (observations) y_t according to the *measurement model*,

$$y_t \sim p_\theta(y_t|x_t). \quad (2.8)$$

The observation process $\{y_t\}$ is assumed to be conditionally independent of the state process $\{x_t\}$, i.e.,

$$p_\theta(y_t|x_1, \dots, x_N) = p_\theta(y_t|x_t), \quad \forall t, 1 \leq t \leq N. \quad (2.9)$$

Furthermore, the observations are assumed to be mutually independent over time,

$$\begin{aligned} p_\theta(y_t, \dots, y_N|x_t, \dots, x_N) &= \prod_{i=t}^N p_\theta(y_i|x_t, \dots, x_N) \\ &= \prod_{i=t}^N p_\theta(y_i|x_i), \quad \forall t, 1 \leq t \leq N. \end{aligned} \quad (2.10)$$

where (2.9) is used to obtain the last equality. In certain tasks, such as convergence proofs, more advanced tools from measure theory (Chung, 1974, Billingsly, 1995) might be needed. This implies that the model has to be defined within a measure theoretic framework. We will not be concerned with measure theory in this thesis, but the interested reader can consult, e.g., Crisan (2001), Crisan and Doucet (2002) for discussions of this kind. The above discussion is summarized by Model 2, referred to as the *hidden Markov model* (HMM) (Doucet et al., 2000a).

Model 2 (Hidden Markov Model (HMM))

The hidden Markov model is defined by

$$x_{t+1} \sim p_\theta(x_{t+1}|x_t), \quad (2.11a)$$

$$y_t \sim p_\theta(y_t|x_t), \quad (2.11b)$$

where θ is used to denote a static parameter.

This model is rather general and in most applications it is sufficient to use one of its special cases. The natural first step in making the class more restrictive is to assume explicit expressions for both the system model and the measurement model, resulting in the state-space model.

2.3 State-Space Models

A state-space model is a model where the relationship between the input signal, the output signal and the noises is provided by a system of first-order differential (or difference) equations. The state vector x_t contain all information there is to know about the system up to and including time t , which is needed to determine the future behavior of the system, given the input. Furthermore, state-space models constitute a very important special case of Model 1, widely studied within the areas of signal processing and systems and control theory. The rest of this section is concerned with various important state-space models, starting with the most general.

2.3.1 Nonlinear State-Space Models

The aim of this section is to provide an introduction to nonlinear, non-Gaussian state-space models. It will also be illustrated that the resulting model is indeed a discrete-time special case of Model 1. The assumption of explicit expressions for both the system model and measurement model in (2.11) result in

$$x_{t+1} = f(x_t, w_t, \theta, t), \quad (2.12a)$$

$$y_t = h(x_t, e_t, \theta, t), \quad (2.12b)$$

where w_t and e_t are independent random variables, commonly referred to as the *process noise* and the *measurement noise*, respectively. The functions f and h in (2.12) describe the evolution of the state variables and the measurements over time. The model is usually restricted even further by assuming that the noise processes enter additively.

Model 3 (Nonlinear state-space model with additive noise)

The nonlinear, discrete-time state-space model with additive noise is given by

$$x_{t+1} = f(x_t, \theta, t) + w_t, \quad (2.13a)$$

$$y_t = h(x_t, \theta, t) + e_t, \quad (2.13b)$$

where w_t and e_t are assumed to be mutually independent noise processes.

Model 3 can be put in the form of Model 2 by the following observation,

$$p_\theta(x_{t+1}|x_t) = p_{w_t}(x_{t+1} - f(x_t, \theta, t)), \quad (2.14a)$$

$$p_\theta(y_t|x_t) = p_{e_t}(y_t - h(x_t, \theta, t)). \quad (2.14b)$$

There are theorems available describing how to obtain similar relations when the noise does not enter additively as in (2.13). For further details on this topic, see Gut (1995), Jazwinski (1970).

The assumption that the observations are mutually independent over time (2.10) translates to mutual independence of the measurement noise e_t over time,

$$p_\theta(y_t, \dots, y_N|x_t, \dots, x_N) = \prod_{i=t}^N p_\theta(y_i|x_i) = \prod_{i=t}^N p_{e_i}(y_i - h(x_i, \theta, i)). \quad (2.15)$$

Furthermore, using conditioning and the Markov property we have

$$p_\theta(x_t, \dots, x_N) = \prod_{i=t}^{N-1} p_\theta(x_{i+1}|x_i) = \prod_{i=t}^{N-1} p_{w_i}(x_{i+1} - f(x_i, \theta, i)). \quad (2.16)$$

Hence, the process noise w_t should also be mutually independent over time. The above discussion does in fact explain how the previous assumptions translate to the use of white noise in Model 3. We could just as well have started from the white noise assumption in Model 3 and motivated the assumptions from this. In the literature the exact definition of white noise differs. Papoulis (1991) refers to *white noise* as a process $\{w_t\}$, which is uncorrelated,

$$\mathbb{E}\{(w_t - \mathbb{E}\{w_t\})(w_s - \mathbb{E}\{w_s\})^T\} = 0, \quad t \neq s. \quad (2.17)$$

A stricter definition is given by Söderström (1994), where independence is required. This is referred to as *strictly* white noise by Papoulis (1991). Furthermore, it is mostly assumed that the mean value of a white noise sequence is zero. We give the following definition.

Definition 2.2 (White noise). A discrete-time stochastic process $\{w_t\}$ is said to be white if it is independent over time, that is

$$p(w_t, w_s) = p(w_t)p(w_s), \quad t \neq s. \quad (2.18)$$

In discussing linear and Gaussian systems it is sufficient to require the process to be uncorrelated according to (2.17), since it is only the two first moments that matter. However, in discussing nonlinear, non-Gaussian systems higher order moments have to be accounted for as well, motivating the independence requirement. Definition 2.2 implies that all the entities of the process $\{w_t\}$ are mutually independent. Hence, there is no information about the future realizations of the white noise process present in the past realizations, implying that white noise is totally unpredictable. The use of white noise can also be motivated from a users perspective. When all systematic information about the studied system has been incorporated in the model equations, there will always remain some random effects which cannot be accounted for. The fact that white noise is totally random, without temporal correlation, implies that it provides a good model for these effects.

In studying the nonlinear system identification problem we will consider a further special case of Model 3. It is a nonlinear state-space model, where the dependence on the static parameters is affine in nature.

Model 4 (Nonlinear state-space model with affine parameters)

A nonlinear state-space model, with affine parameter dependence is defined as

$$x_{t+1} = f_1(x_t, u_t, t)\theta + f_2(x_t, u_t, t) + w_t, \quad (2.19a)$$

$$y_t = h_1(x_t, u_t, t)\theta + h_2(x_t, u_t, t) + e_t, \quad (2.19b)$$

where $w_t \sim \mathcal{N}(0, Q_t)$ and $e_t \sim \mathcal{N}(0, R_t)$ are white noise sequences.

Note that, since this model class will be used for system identification, the known input signals u_t are explicitly included. A key observation worth mentioning is that, conditioned on the nonlinear states x_t this is a rather simple model, where the parameters can be solved for using standard linear regression techniques. This observation is utilized in Paper F. The idea of using conditioning in order to obtain simpler models naturally brings us over to the next section dealing with mixed linear/nonlinear state-space models.

2.3.2 Mixed Linear/Nonlinear State-Space Models

It is a very ambitious endeavor to solve the estimation problems arising when the underlying model is nonlinear. We have tried to approach this problem by studying certain tractable sub-classes of the general nonlinear state-space model. An important part of the thesis is in fact the derivation and application of estimation algorithms especially devised to exploit linear sub-structures inherent in the underlying models. When such a sub-structure is present it is instructive to partition the state variable according to

$$x_t = \begin{pmatrix} x_t^l \\ x_t^n \end{pmatrix}, \quad (2.20)$$

where x_t^l denotes the linear state variables and x_t^n denotes the nonlinear state variables. Models allowing for the partitioning (2.20) will be referred to as *mixed linear/nonlinear state-space models*. When there is a linear sub-structure present in the model we can take advantage of this in deriving algorithms to solve various estimation problems. The most general mixed linear/nonlinear state-space model discussed in this thesis is summarized in Model 5. Note that the possible dependence on unknown static parameters θ has

been suppressed for brevity. For a more thorough discussion regarding this model, see Paper A.

Model 5 (Mixed linear/nonlinear state-space model)

The mixed linear/nonlinear state-space model is given by

$$x_{t+1}^n = f^n(x_t^n, t) + A^n(x_t^n, t)x_t^l + G^n(x_t^n, t)w_t^n, \quad (2.21a)$$

$$x_{t+1}^l = f^l(x_t^n, t) + A^l(x_t^n, t)x_t^l + G^l(x_t^n, t)w_t^l, \quad (2.21b)$$

$$y_t = h(x_t^n, t) + C(x_t^n, t)x_t^l + e_t, \quad (2.21c)$$

where the process noise is assumed white and Gaussian distributed with

$$w_t = \begin{pmatrix} w_t^l \\ w_t^n \end{pmatrix} \sim \mathcal{N}(0, Q_t), \quad Q_t = \begin{pmatrix} Q_t^l & Q_t^{ln} \\ (Q_t^{ln})^T & Q_t^n \end{pmatrix}. \quad (2.22a)$$

The measurement noise is assumed white and Gaussian distributed $e_t \sim \mathcal{N}(0, R_t)$. Furthermore, x_0^l is Gaussian distributed $x_0^l \sim \mathcal{N}(\bar{x}_0, \bar{P}_0)$. The density of x_0^n can be arbitrary, but it is assumed known.

Conditioned on the nonlinear states, the model described above is linear². This can be used in deriving estimation algorithms for models of this type. An interesting algorithm for this is the *marginalized particle filter* or the Rao-Blackwellized particle filter (Doucet et al., 2000a). It is briefly introduced in Section 4.4 and thoroughly treated in Paper A. Model 5 is quite general and in most applications it is sufficient to consider a special case of it. A quite common and important special case is when the dynamics is linear and the measurement equation is nonlinear.

Model 6 (Model 5 with linear dynamics and nonlinear measurements)

A common special case of Model 5 occurs when the dynamics is linear and the measurements are nonlinear.

$$x_{t+1}^n = A_{n,t}^n x_t^n + A_{l,t}^n x_t^l + G_t^n w_t^n, \quad (2.23a)$$

$$x_{t+1}^l = A_{n,t}^l x_t^n + A_{l,t}^l x_t^l + G_t^l w_t^l, \quad (2.23b)$$

$$y_t = h(x_t^n, t) + e_t, \quad (2.23c)$$

where $w_t^n \sim \mathcal{N}(0, Q_t^n)$ and $w_t^l \sim \mathcal{N}(0, Q_t^l)$. The distribution for e_t can be arbitrary, but it is assumed known.

In positioning and target tracking applications models of this type are quite commonly used. Several examples of this are given in Paper H and the references therein. For more information concerning various modeling issues, see, e.g., Gustafsson et al. (2002), Bar-Shalom and Li (1993), Li and Jilkov (2001, 2003).

²Strictly speaking the model is affine, due to the possible presence of the term f^l .

2.3.3 Linear State-Space Models

The most important special case of Model 3 is probably the linear (f and h are linear functions) state-space model, subject to Gaussian noise. The reason for this is probably the fundamental work of Kalman in the 1960s on the prediction and linear quadratic control, based on this model.

Model 7 (Linear state-space model with Gaussian noise)

The discrete-time linear state-space model, subject to Gaussian noise is given by

$$x_{t+1} = A_t(\theta)x_t + w_t, \quad (2.24a)$$

$$y_t = C_t(\theta)x_t + e_t, \quad (2.24b)$$

where $w_t \sim \mathcal{N}(0, Q_t(\theta))$, $e_t \sim \mathcal{N}(0, R_t(\theta))$, and $\mathbb{E}\{w_t e_t^T\} = 0$.

In Model 7 above, δ_{ts} is the Kronecker delta function, which is 0 whenever $t \neq s$, and 1, when $t = s$. It is important to note that Model 7 is a bit more general than it seems at a first glance. The reason is that if we have colored noise processes or a non-zero cross-correlation between w_t and e_t the model can be rewritten in the form (2.24). For details, see Kailath et al. (2000).

The theory concerning linear state-space models is by now quite mature. For the details concerning linear system theory two good references are Rugh (1996) and Kailath (1980). For the linear state estimation problem Kailath et al. (2000) is the standard reference. The parameter estimation problem is thoroughly treated in Ljung (1999), Söderström and Stoica (1989).

2.4 Linear Differential-Algebraic Equations

In the thesis, Model 3 and some of its special cases are used extensively. However, we will also discuss possible extensions in terms of differential-algebraic equations. The first obstacle to overcome is to solve the problem of introducing stochastic processes into this type of model. This is not as simple as it is with state-space models. In this section the problem is briefly described and in Paper C a detailed proposal for how to solve this problem is provided. These results have recently been refined and sharpened, see Gerdin et al. (2005a,b). The *linear stochastic differential-algebraic equation* is defined in Model 8 below.

Model 8 (Linear stochastic differential-algebraic equation model)

The linear stochastic differential-algebraic equation model is given by

$$E(\theta)\dot{z}(t) + F(\theta)z(t) = B_w(\theta)w(t), \quad (2.25a)$$

$$y(t_k) = C(\theta)z(t_k) + e(t_k), \quad (2.25b)$$

where $E(\theta)$ might be singular and $w(t)$ and $e(t_k)$ are white Gaussian noises.

The reason for incorporating white noise in linear DAEs is that it opens up for using the standard methods of statistical signal processing. More specifically, it allows for a

systematic treatment of the two problems of estimating the internal variables $z(t)$ and static parameters θ . The system identification problem is thoroughly treated in Gerdin (2004) and Gerdin et al. (2005b) and estimation of the internal variables is discussed in Paper C and Gerdin et al. (2005b). In the discrete-time case much has already been done, see, e.g., Dai (1987, 1989), Darouach et al. (1993), Deng and Liu (1999), Nikoukhah et al. (1998, 1999). However, models obtained from object-oriented modeling languages are mostly in continuous-time, further motivating the need to be able to introduce stochastic processes in continuous-time DAE models.

The problem of introducing stochastic processes in linear differential-algebraic equations boils down to making sure that the implicit differentiation of w that may be hidden in the equations does not lead to intractable mathematical objects, such as differentiated white noise. In order to understand this it is instructive to rewrite the equations in the standard form provided by Theorem 2.1.

Theorem 2.1 (Standard form for Model 8)

Suppose that there exists a scalar λ such that $\lambda E + F$ is invertible. Then there exist nonsingular matrices P and Q such that the transformation

$$PEQQ^{-1}\dot{z}(t) + PFQQ^{-1}z(t) = PB_w w(t), \quad (2.26)$$

allows us to write (2.25) as

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} + \begin{pmatrix} -A & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} w(t), \quad (2.27)$$

where N is a matrix of nilpotency k , i.e., $N^k = 0$ for some k . (Q is used as a variable substitution, $x(t) = Q^{-1}z(t)$ and P is multiplied from the left in (2.25a).)

Proof: Kronecker's canonical form (see Kailath, 1980, Gantmacher, 1959) provides a proof for the existence of this standard form. For a detailed proof see Gerdin (2004). \square

It is worth noting that although this standard form always exists it can indeed be numerically hard to find the transformation matrices P and Q . However, using the ideas from Varga (1992) this problem can be handled, see, e.g., Gerdin (2004), Gerdin et al. (2005b) for details regarding these numerical issues. If (2.25) is rewritten according

$$\dot{x}_1(t) = Ax_1(t) + G_1w(t), \quad (2.28a)$$

$$x_2(t) = \sum_{i=0}^{k-1} (-N)^i G_2 \frac{d^i w(t)}{dt^i}, \quad (2.28b)$$

it can be seen that white noise is prevented from being differentiated if

$$NG_2 = 0. \quad (2.29)$$

In Paper C this is utilized to derive conditions on the model class that imply that white noise is not differentiated.

3

Nonlinear State Estimation

RECURSIVE nonlinear state estimation theory is the topic of the present chapter. As previously mentioned, the state estimation problem is addressed mainly within a probabilistic framework. More specifically, the approach is heavily influenced by the Bayesian view of estimation. This implies that the complete solution to the estimation problem is provided by the probability density function $p(x_t|Y_s)$. This density function contains all available information about the state variable. Depending on the relation between t and s in $p(x_t|Y_s)$ three different estimation problems are obtained

- The *filtering* problem, $t = s$.
- The *prediction* problem, $t > s$.
- The *smoothing* problem, $t < s$.

This chapter will illustrate how the expressive power of the probability density functions opens up for a rather systematic treatment of the three problems mentioned above. When a representation for $p(x_t|Y_s)$ is obtained it can be used to estimate the expected value of any function g of the state variables, $I(g(x_t))$ according to

$$I(g(x_t)) \triangleq E_{p(x_t|Y_s)} \{g(x_t)\} = \int_{\mathbf{R}^{n_x}} g(x_t) p(x_t|Y_s) dx_t. \quad (3.1)$$

The chapter starts with a brief history of the estimation problem in Section 3.1. In Section 3.2 the general solutions to the filtering, prediction and smoothing problems are derived, in terms of probability density functions. The discussion then continues with Section 3.3, where several of the most common estimates (3.1) are introduced. The state estimation problem arising from nonlinear systems is discussed in Section 3.4. The common special case of linear models, subject to Gaussian noise is then treated in Section 3.5. Change detection can be used to adapt the models according to changes in the underlying

system, with better state estimates as result. This is the topic of Section 3.6. Finally, the chapter is concluded with Section 3.7, where we provide a deterministic view of the estimation problem and illustrate how this together with convex optimization techniques can be used to handle constraints present in the problem.

3.1 Brief History of the State Estimation Problem

The aim of this section is to provide a short historic account of the estimation problem. We will merely skim the surface of this fascinating topic, but we will try to provide adequate references for further studies. Some general references are Spall (1988), Jazwinski (1970), Sorenson (1970), Mendel and Giesecking (1971).

The first attempts to systematically approach the estimation problem, as it is known today, were taken by Gauss and Legendre in studying astronomical problems during the late 18th and the early 19th century. More specifically, they tried to estimate the positions of planets and comets using telescopic measurements. Gauss made use of the method of least-squares for the first time in 1795 at the age of 18. However, it was not until 1809 that he published his results in his book *Theoria Motus Corporum Celestium* (Gauss, 1809). A few years earlier, in 1805 Legendre had independently invented and published the method in his book *Nouvelles méthodes pour la détermination des orbites des comètes*. This gave rise to a big dispute between Gauss and Legendre, concerning who was the inventor of the least-squares method (Sorenson, 1970). A thorough discussion of the early contributions to estimation theory is provided by Seal (1967) and Sorenson (1970).

The next major development in the study of the estimation problem came in the 1940s, with the filtering work of Wiener (1949) and Kolmogorov. They both studied the problem of extracting an interesting signal in a signal-plus-noise setting and independently solved the problem, using a linear minimum mean-square technique. The solution is based on the rather restrictive assumptions of access to an infinite amount of data and that all involved signals can be described as stationary stochastic processes. During the 1940s and the 1950s much research was directed towards trying to relax those assumptions and extend the Wiener – Kolmogorov filtering theory. The breakthrough came with the Kalman filter, introduced by Kalman (1960)¹. It changed the conventional formulation of the estimation problem and in doing so it moved the research into a completely new direction, away from the theory of stationary stochastic processes. The key ingredient in this turn was the Kalman filter’s inherent access to the powerful state-space theory, that had recently been developed within the automatic control community. The important connection between the estimation problem and the state-space theory had now been established.

The Kalman filter allows us to drop the assumptions of stationary signals and access to an infinite amount of data. Furthermore, Kalman’s state-space approach naturally lends itself to multivariable problems, whereas the Wiener – Kolmogorov theory and other frequency domain techniques bump into severe problems when the extension to the multivariable case is considered.

During the 1960s, 1970s and the 1980s many suggestions were given on how to

¹In the late 1800s, the Danish astronomer T. N. Thiele developed a recursive procedure, for determining the distance from Copenhagen to Lund. Interestingly enough his solution was a special case of the Kalman filter (Spall, 1988). A modern discussion of Thiele’s work is provided by Lauritzen (1981) and Hald (1981).

extend the Kalman filtering theory to handle more general estimation problems. In 1993 the *particle filter* was first introduced by Gordon et al. (1993). It provides a systematic procedure for solving the nonlinear, non-Gaussian estimation problem. As Kailath (1974) points out the Kalman filter was the new idea that allowed the field to move in a new, fruitful direction after the Wiener – Kolmogorov theory. Perhaps we can think of the particle filter along the same line, as a new, fruitful direction allowing us to tackle even harder estimation problems.

3.2 Conceptual Solution

This section is concerned with the problem of calculating the probability density functions relevant in solving the estimation problem. The discussion will be rather general using Model 2 defined in Section 2.3.1, briefly summarized in (3.2) for convenience

$$x_{t+1} \sim p(x_{t+1}|x_t), \quad (3.2a)$$

$$y_t \sim p(y_t|x_t). \quad (3.2b)$$

In the development that follows Bayes' theorem and the Markov property will be instrumental. The Markov property was previously defined in Definition 2.1. Using the two stochastic variables x and y , Bayes' theorem for probability density functions is given by

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y, x)}{p(y)}. \quad (3.3)$$

Consider the filtering density,

$$\begin{aligned} p(x_t|Y_t) &= p(x_t|y_t, Y_{t-1}) = \frac{p(y_t|x_t, Y_{t-1})p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \\ &= \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})}, \end{aligned} \quad (3.4)$$

where $p(y_t|Y_{t-1})$ can be calculated according to

$$\begin{aligned} p(y_t|Y_{t-1}) &= \int_{\mathbf{R}^{n_x}} p(y_t, x_t|Y_{t-1}) dx_t = \int_{\mathbf{R}^{n_x}} p(y_t|x_t, Y_{t-1})p(x_t|Y_{t-1}) dx_t \\ &= \int_{\mathbf{R}^{n_x}} p(y_t|x_t)p(x_t|Y_{t-1}) dx_t. \end{aligned} \quad (3.5)$$

Furthermore, in order to derive the expression for the one step ahead prediction density $p(x_{t+1}|Y_t)$ the following equation is integrated w.r.t. x_t ,

$$p(x_{t+1}, x_t|Y_t) = p(x_{t+1}|x_t, Y_t)p(x_t|Y_t) = p(x_{t+1}|x_t)p(x_t|Y_t), \quad (3.6)$$

resulting in the following expression

$$p(x_{t+1}|Y_t) = \int_{\mathbf{R}^{n_x}} p(x_{t+1}|x_t)p(x_t|Y_t) dx_t. \quad (3.7)$$

This equation is commonly referred to as the *Chapman–Kolmogorov* equation (Jazwinski, 1970). It is straightforward to generalize this idea to obtain an expression for the k -step ahead prediction density. Rather than integrating $p(x_{t+1}, x_t | Y_t)$ w.r.t. x_t we integrate $p(x_{t+k}, \dots, x_t | Y_t)$ w.r.t. $X_{t:t+k-1} = \{x_i\}_{i=t}^{t+k-1}$. Hence,

$$\begin{aligned} p(x_{t+k} | Y_t) &= \int_{\mathbf{R}^{kn_x}} p(x_{t+k}, \dots, x_t | Y_t) dx_{t:t+k-1} \\ &= \int_{\mathbf{R}^{kn_x}} \prod_{i=1}^k p(x_{t+i} | x_{t+i-1}) p(x_t | Y_t) dx_{t:t+k-1}. \end{aligned} \quad (3.8)$$

In deriving suitable expressions for the smoothing density several alternatives exist. Let us first derive an expression for the marginal smoothing density $p(x_t | Y_N)$ by observing that

$$p(x_t | Y_N) = \int_{\mathbf{R}^{n_x}} p(x_t, x_{t+1} | Y_N) dx_{t+1}, \quad (3.9)$$

where

$$p(x_t, x_{t+1} | Y_N) = p(x_t | x_{t+1}, Y_N) p(x_{t+1} | Y_N). \quad (3.10)$$

Furthermore,

$$\begin{aligned} p(x_t | x_{t+1}, Y_N) &= p(x_t | x_{t+1}, Y_t, Y_{t+1:N}) \\ &= \frac{p(Y_{t+1:N} | x_t, x_{t+1}, Y_t) p(x_t | x_{t+1}, Y_t)}{p(Y_{t+1:N} | x_{t+1}, Y_t)} = p(x_t | x_{t+1}, Y_t), \end{aligned} \quad (3.11)$$

where the last equality follows from the fact that given x_{t+1} , there is no further information about $Y_{t+1:N}$ available in x_t . Using this result the smoothing density (3.9) can be written according to

$$\begin{aligned} p(x_t | Y_N) &= \int_{\mathbf{R}^{n_x}} p(x_t | x_{t+1}, Y_t) p(x_{t+1} | Y_N) dx_{t+1} \\ &= \int_{\mathbf{R}^{n_x}} \frac{p(x_{t+1} | x_t, Y_t) p(x_t | Y_t)}{p(x_{t+1} | Y_t)} p(x_{t+1} | Y_N) dx_{t+1} \\ &= p(x_t | Y_t) \int_{\mathbf{R}^{n_x}} \frac{p(x_{t+1} | x_t) p(x_{t+1} | Y_N)}{p(x_{t+1} | Y_t)} dx_{t+1}. \end{aligned} \quad (3.12)$$

Another useful expression for the smoothing density is referred to as the *two-filter formula*. See Kitagawa (1994), Bresler (1986) for a detailed treatment of this formula.

Similar derivations to the ones given above can be found for instance in Ho and Lee (1964), Jazwinski (1970), Kitagawa (1991). For future reference the main results are collected in Theorem 3.1.

Theorem 3.1

If the dynamic model is given by (3.2) the filter density $p(x_t|Y_t)$, the one step ahead density $p(x_{t+1}|Y_t)$, and the marginal smoothing density $p(x_t|Y_N)$ are given by

$$p(x_t|Y_t) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})}, \quad (3.13a)$$

$$p(x_{t+1}|Y_t) = \int_{\mathbf{R}^{n_x}} p(x_{t+1}|x_t)p(x_t|Y_t) dx_t, \quad (3.13b)$$

$$p(x_t|Y_N) = p(x_t|Y_t) \int_{\mathbf{R}^{n_x}} \frac{p(x_{t+1}|x_t)p(x_{t+1}|Y_N)}{p(x_{t+1}|Y_t)} dx_{t+1}, \quad (3.13c)$$

where

$$p(y_t|Y_{t-1}) = \int_{\mathbf{R}^{n_x}} p(y_t|x_t)p(x_t|Y_{t-1}) dx_t. \quad (3.13d)$$

Given the complexity of the problem it is actually quite remarkable that we are able to derive a result as the one given in Theorem 3.1 above. However, there is a severe problem with this solution, the multidimensional integrals involved only permit an analytical solution in a few special cases. The most important special case is when the dynamic model is linear and the involved stochastic variables are normal, which has been extensively discussed in the literature over the last decades. This is due to the fact that the mathematics involved is tractable, but most importantly it hinges on the fact that there are a vast amount of real world applications where this special case has been successfully applied. However, most applications would perform better if the nonlinear estimation problem could be properly solved. This would also allow us to tackle more complicated applications, which do not lend themselves to linear algorithms.

3.3 Point Estimates

The task of finding a point estimate can, in abstract terms, be cast as a problem of finding a transformation m_t , which makes use of the information in the measurements and the known input signals to produce estimates of the states of interest.

$$m_t : U_s \times Y_s \rightarrow \mathbf{R}^{n_x} \quad (3.14)$$

All information available in the measurements has been processed and inferred into the density function $p(x_t|Y_s)$. This density function can then be used to derive various point estimates, which is normally what the user would expect from the estimation algorithm. Typically, the application does not need the entire probability density function. Instead it needs to know how the values of the various states evolve over time and it also need a quality assessment of these values. It is reasonable to claim that an estimate is useless, if we do not know how good it is. Since a probabilistic framework is employed, this opens up for using the tools available in probability theory and statistics for assessing the quality of estimates, such as covariances, confidence regions, tests, etc.

This section is concerned with some of the most common mappings (3.14) present in the literature. Most of the estimates are indeed based on approximations of the probability density functions $p(x_t|Y_s)$, but the estimates can also be based on deterministic considerations. This approach to estimation is discussed in Section 3.7. For more information about various estimates, see, e.g., Kailath et al. (2000), Jazwinski (1970), Kay (1993), Anderson and Moore (1979).

From a probabilistic point of view a rather appealing point estimate is provided by choosing the value that minimizes the variance of the estimation error, referred to as the *minimum variance* (MV) estimate

$$\hat{x}^{MV} \triangleq \arg \min_{\hat{x}} \mathbb{E} \left\{ \|x - \hat{x}\|^2 | y \right\} \quad (3.15)$$

where $\|x\|^2 = x^T x$. It is in fact possible to derive an explicit expression for this estimate.

$$\begin{aligned} \mathbb{E} \left\{ \|\hat{x} - x\|^2 | y \right\} &= \mathbb{E} \left\{ (x - \hat{x})^T (x - \hat{x}) | y \right\} \\ &= \mathbb{E} \{x^T x | y\} - 2\hat{x}^T \mathbb{E} \{x | y\} + \hat{x}^T \hat{x} \\ &= \|\hat{x} - \mathbb{E} \{x | y\}\|^2 + \mathbb{E} \{\|x\|^2 | y\} - \|\mathbb{E} \{x | y\}\|^2 \end{aligned} \quad (3.16)$$

The two last terms in (3.16) are independent of \hat{x} and (3.16) is clearly minimized by

$$\hat{x}^{MV} = \mathbb{E} \{x | y\} = \int x p(x | y) dx. \quad (3.17)$$

The above calculation explains the name, *minimum mean square error* (MMSE), which is commonly used as an alternative name for the estimate (3.17).

Another point estimate which suggests itself, within the probabilistic framework, is the most probable outcome,

$$\hat{x}^{MAP} \triangleq \arg \max_x p(x | y) = \arg \max_x p(y | x) p(x), \quad (3.18)$$

which is referred to as the *maximum a posteriori* (MAP) estimate. In the second equality of (3.18) Bayes' theorem is employed, together with the fact that the maximization is performed over x . The *prior* density function $p(x)$ in (3.18) is within the classical school assumed completely uninformative, giving rise to the *maximum likelihood* (ML) estimate,

$$\hat{x}^{ML} \triangleq \arg \max_x p(y | x). \quad (3.19)$$

The method of maximum likelihood was introduced by Fisher (1912, 1922). The maximum likelihood method is used extensively in the study of a certain class of nonlinear system identification problems, see Paper F.

3.4 Nonlinear Systems

Most of the problems encountered in practice are of a nonlinear nature, which implies that we have to be able to solve estimation problems in the context of nonlinear systems.

The nonlinear systems theory is, as opposed to its linear counterpart, far from mature. However, there is a flurry of results readily available, see, e.g., the monographs by Khalil (2002) and Isidori (1989). When it comes to nonlinear estimation theory the book by Jazwinski (1970) is still very interesting reading.

There is a wealth of representations available when it comes to nonlinear systems. However, the most common representation, at least when it comes to solving estimation problems is given by Model 3, repeated here for convenience

$$x_{t+1} = f(x_t, t) + w_t, \quad w_t \sim \mathcal{N}(0, Q_t), \quad (3.20a)$$

$$y_t = h(x_t, t) + e_t, \quad e_t \sim \mathcal{N}(0, R_t). \quad (3.20b)$$

In discussing the implications of Theorem 3.1 we observed that, in general, there does not exist any analytical solution to the nonlinear recursive estimation problem. This implies that we are forced to approximations of some kind in order to approach this problem. The approximations suggested in literature this far, can roughly be divided into two different classes, local and global. This distinction has previously been discussed, for instance by Sorenson (1974) and Kulhavý (1996). The local approach approximates (3.20) using a locally valid linear, Gaussian model. This is then used in conjunction with the Kalman filter to obtain the estimates. The idea underpinning the global approach is indeed more appealing. It makes use of the nonlinear model and tries to approximate the solution provided in Theorem 3.1. Hence, it is a matter of either approximating the model and using the linear, Gaussian estimator or using the correct model and approximate the optimal solution. Despite the fact that there are a lot of different nonlinear estimators available, the local approach is still the most commonly used nonlinear estimator when it comes to applications. This approach is explained in more detail in the subsequent section. However, in recent years the sequential Monte Carlo methods have emerged as interesting global approaches, gaining more and more ground, both when it comes to theory and when it comes to applications.

3.4.1 Local Approximations

The idea employed in local methods is to approximate the nonlinear model by a linear, Gaussian model. This model is only valid locally, but the Kalman filter can readily be applied. The first approach along those lines was to linearize the model along a nominal trajectory, resulting in the *linearized Kalman filter* (Kailath et al., 2000). An improvement to this was suggested by S. F. Schmidt et.al. They suggested that the linearization should be performed around the current estimate, rather than around a nominal trajectory. The result is the *extended Kalman filter* (or more appropriately the Schmidt EKF) (Smith et al., 1962, Schmidt, 1966). To the best of the authors knowledge the paper by Smith et al. (1962) describes the first practical application of the (extended) Kalman filter. More specifically, the local approximation is obtained by linearizing the nonlinear model (3.20)

by applying a first-order Taylor expansion around the current estimate,

$$f(x_t, t) \approx f(\hat{x}_{t|t}, t) + \frac{\partial f(x, t)}{\partial x} \Big|_{x=\hat{x}_{t|t}} (x_t - \hat{x}_{t|t}), \quad (3.21a)$$

$$h(x_t, t) \approx h(\hat{x}_{t|t-1}, t) + \frac{\partial h(x, t)}{\partial x} \Big|_{x=\hat{x}_{t|t-1}} (x_t - \hat{x}_{t|t-1}). \quad (3.21b)$$

Using this approximation in (3.20) gives

$$x_{t+1} = f(\hat{x}_{t|t}, t) - F_t \hat{x}_{t|t} + F_t x_t + w_t, \quad (3.22a)$$

$$y_t = h(\hat{x}_{t|t-1}, t) - H_t \hat{x}_{t|t-1} + H_t x_t + e_t, \quad (3.22b)$$

where

$$F_t \triangleq \frac{\partial f(x, t)}{\partial x} \Big|_{x=\hat{x}_{t|t}}, \quad H_t \triangleq \frac{\partial h(x, t)}{\partial x} \Big|_{x=\hat{x}_{t|t-1}}. \quad (3.23)$$

The approximate model given in (3.22) is a linear, Gaussian model in x_t , which implies that the Kalman filter given in Corollary 3.1 can be applied. The result is the extended Kalman filter, given in Algorithm 3.1.

Algorithm 3.1 (Extended Kalman Filter (EKF))

Consider Model 3, repeated in (3.20). An approximate sub-optimal estimate for the filter density function $p(x_t | Y_t)$, obtained by linearization, is recursively given according to

$$\hat{p}(x_t | Y_t) = \mathcal{N}(x | \hat{x}_{t|t}, P_{t|t}), \quad (3.24a)$$

$$\hat{p}(x_{t+1} | Y_t) = \mathcal{N}(x | \hat{x}_{t+1|t}, P_{t+1|t}), \quad (3.24b)$$

where

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t (y_t - h(\hat{x}_{t|t-1}, t)), \quad (3.25a)$$

$$P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}, \quad (3.25b)$$

$$\hat{x}_{t+1|t} = f(\hat{x}_{t|t}, t), \quad (3.25c)$$

$$P_{t+1|t} = F_t P_{t|t} F_t^T + Q_t, \quad (3.25d)$$

$$K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1}, \quad (3.25e)$$

with initial values $\hat{x}_{1|0} = \bar{x}_1$ and $P_{1|0} = \bar{\Pi}_1$. Furthermore, F_t and H_t are defined by

$$F_t = \frac{\partial f(x, t)}{\partial x} \Big|_{x=\hat{x}_{t|t}}, \quad H_t = \frac{\partial h(x, t)}{\partial x} \Big|_{x=\hat{x}_{t|t-1}}. \quad (3.26)$$

For a more thorough treatment of the EKF the reader is referred to Jazwinski (1970), Anderson and Moore (1979), Kailath et al. (2000). An application focused discussion is given in Sorenson (1985). One of the problems inherent in the EKF is that it might diverge. The literature contains several more or less *ad hoc* methods trying to counteract this phenomenon and to further enhance the general performance of the EKF. To mention a few examples we have, the *iterated EKF* treated by Kailath et al. (2000) and higher-order Taylor expansions discussed by Bar-Shalom and Fortmann (1988) and Gustafsson (2000).

3.4.2 Global Approximations

The solution to the nonlinear recursive estimation problem is given by Theorem 3.1. This fact is neglected by methods based on local model approximations. However, if we choose to use this theorem the nonlinear models derived from the underlying physics can be used and rather than approximating the models, the optimal solution is approximated using numerical methods. Over the years several different methods for performing this approximation have appeared. These methods are of two different kinds, either the probability density functions of interest are parameterized by a finite number of parameters, which are updated according to Theorem 3.1 or the integrals in Theorem 3.1 are handled using numerical integration. Here, only a few of the most important global approximations are mentioned. For more references on this topic see, e.g., Kulhavý (1996), Bergman (1999), Sorenson (1974).

One of the first approaches using an approximation based on a finite set of parameters is the *Gaussian sum* approach by Sorenson and Alspach (1971), Alspach and Sorenson (1972), where the filtering density is approximated using a sum of Gaussian densities according to

$$p(x_t|Y_t) \approx \sum_{i=1}^N q_t^{(i)} \mathcal{N}\left(x | \hat{x}_{t|t}^{(i)}, P_{t|t}^{(i)}\right), \quad \sum_{i=1}^N q_t^{(i)} = 1, \quad q_t^{(i)} \geq 0, \quad \forall i. \quad (3.27)$$

Another approximation is provided by the *point-mass filter* originally suggested by Bucy and Senne (1971) which, as the name reveals, approximates the filtering density by a set of points on a predefined grid,

$$p(x_t|Y_t) \approx \sum_{i=1}^N q_t^{(i)} \delta\left(x_t - x_t^{(i)}\right), \quad \sum_{i=1}^N q_t^{(i)} = 1, \quad q_t^{(i)} \geq 0, \quad \forall i. \quad (3.28)$$

This idea has been refined and generalized over the years using for instance piecewise constant approximations and spline interpolations. The point-mass filter is thoroughly treated in Bergman (1999), Bergman et al. (1999), where it is also applied to the aircraft navigation problem. Another approach which recently has appeared is the *unscented Kalman filter* (UKF), which is based on the unscented transform, discussed in Julier et al. (2000), Julier and Uhlmann (2004). The basic idea here is to use a set of grid points in the state-space, chosen by the unscented transform.

There is another family of algorithms which makes use of multiple models in order to derive an estimate. They use a set of models describing various behaviors of the underlying system. This approach is common in target tracking applications, where different maneuvers of the tracked vehicle constitutes the different models. Examples of algorithms of this type are the *interacting multiple model* (IMM) and the *generalized pseudo-Bayesian* (GPB) approaches, which are thoroughly described by Bar-Shalom and Li (1993), with the target tracking application in mind. Yet another algorithm within this family is the *range parameterized extended Kalman filter* (RPEKF) (Peach, 1995, Arulampalam and Ristic, 2000), which is described and applied to a bearings-only tracking application by Karlsson (2005).

Another approach, which can be interpreted as an extension of the point-mass filter is provided by the *sequential Monte Carlo methods*, referred to as the *particle filter* (Gordon

et al., 1993, Kitagawa, 1996, Doucet et al., 2001a) in the filtering case. In these algorithms the probability density function is also approximated by a set of grid points. However, the grid is not chosen deterministically, as is the case in point-mass filters. Due to its relevance for the present thesis the sequential Monte Carlo methods are discussed in more detail in Chapter 4. It is worth mentioning that there is a vast amount of literature dealing with different combinations and variations of the approaches discussed above.

3.5 Linear Systems

The classic special case when it comes to estimation, and systems theory in general, is constituted by linear systems subject to Gaussian noise processes. The theory concerned with linear systems is by now rather mature, see, e.g., Rugh (1996), Kailath (1980) for a general treatment without stochastic processes. The linear dynamic model was introduced as Model 7 in Section 2.3.3, but the equations, including a known input signal u_t , are repeated here for convenience,

$$x_{t+1} = A_t x_t + B_t u_t + w_t, \quad w_t \sim \mathcal{N}(0, Q_t), \quad (3.29a)$$

$$y_t = C_t x_t + D_t u_t + e_t, \quad e_t \sim \mathcal{N}(0, R_t). \quad (3.29b)$$

A solid treatment of the linear estimation problem is given by Kailath et al. (2000), the fundamental innovation process is extensively used. In understanding linear estimation it is advantageous to appeal to the geometrical intuition, which is possible due to the fact that linear estimation can be interpreted as projections in Hilbert spaces. There exist a vast amount of literature dealing with the linear estimation problem, and the Kalman filter in particular, see, e.g., Kailath et al. (2000), Kay (1993), Jazwinski (1970), Anderson and Moore (1979), Sorenson (1985), Gustafsson (2000), West and Harrison (1997), Harvey (1989), Bryson and Ho (1975).

An important property of the linear model (3.29) is that all density functions involved are Gaussian. This is due to the fact that a linear transformation of a Gaussian random variable will result in a new Gaussian random variable. Furthermore, a Gaussian density function is completely parameterized by two parameters, the first and second order moments, i.e., the mean and the covariance. This implies that if it is assumed that the underlying model is given by (3.29) the recursions in Theorem 3.1 can be recast as recursive relations for the mean values and the covariances of the involved probability density functions. In Section 3.5.1 this is illustrated for the filtering and the prediction densities, which will result in an important corollary to Theorem 3.1. A second corollary is given in Section 3.5.2, where the smoothing problem is considered.

3.5.1 Filtering and Prediction

The special case obtained by assuming a linear, Gaussian model (3.29) allows for an explicit solution to the expressions given in Theorem 3.1. The filtering and one-step ahead prediction solutions are given by the Kalman filter, first derived by Kalman (1960) and Kalman and Bucy (1961). Before stating the theorem the notation $\hat{x}_{t|s}$ is introduced, which denotes the estimate of the state x at time t using the information available in the measurements up to and including time s . In other words, $\hat{x}_{t|s} = E\{x_t|Y_s\}$.

Corollary 3.1 (Kalman filter)

Consider (3.29) and assume that the initial state is distributed as $x_0 \sim \mathcal{N}(\bar{x}_0, \bar{P}_0)$. Then, the estimates for the filtering density function and the one step ahead prediction density function are both normal, according to

$$\hat{p}(x_t|Y_t) = \mathcal{N}(x | \hat{x}_{t|t}, P_{t|t}), \quad (3.30a)$$

$$\hat{p}(x_{t+1}|Y_t) = \mathcal{N}(x | \hat{x}_{t+1|t}, P_{t+1|t})), \quad (3.30b)$$

where

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - C_t\hat{x}_{t|t-1} - D_t u_t), \quad (3.31a)$$

$$P_{t|t} = P_{t|t-1} - K_t C_t P_{t|t-1}, \quad (3.31b)$$

$$\hat{x}_{t+1|t} = A_t \hat{x}_{t|t} + B_t u_t, \quad (3.31c)$$

$$P_{t+1|t} = A_t P_{t|t} A_t^T + Q_t, \quad (3.31d)$$

$$K_t = P_{t|t-1} C_t^T (C_t P_{t|t-1} C_t^T + R_t)^{-1}, \quad (3.31e)$$

with initial values $\hat{x}_{0|-1} = \bar{x}_0$ and $P_{0|-1} = \bar{P}_0$.

Proof: There are many different ways in which this result can be proved. In Appendix A a proof based on the results of Theorem 3.1 is provided. More specifically, the relevant expressions from Theorem 3.1 are simplified using the imposed linear, Gaussian model (3.29). These calculations can also be found in Ho and Lee (1964), Nordlund (2002). For alternative proofs, see, e.g., Kailath et al. (2000), Anderson and Moore (1979), Gustafsson (2000). An interesting proof is given by Rao (2000), where the Kalman filter is obtained as the recursive solution to a weighted least-squares problem. \square

The intuition for the Kalman filter is helped by thinking in terms of time updates and measurement updates. The *measurement update* is given in (3.31a) – (3.31b) and the name derives from the fact that these are the equations where the information in the present measurement y_t is incorporated into the estimate. In (3.31a) this implies that the state estimate is adjusted as a weighted average of the previous estimate and the new information available in y_t . The uncertainty is reduced in (3.31b) as a direct consequence of the fact that new information has been added. Furthermore, the *time update* corresponds to a prediction, implying an increased uncertainty (3.31d). Due to the fact that the process noise w_t , by definition, cannot be predicted the state evolution is obtained simply by using the deterministic part of the dynamic model, as in (3.31c).

An important, if not the most important, factor in making the Kalman filter so fundamental is its applicability. The first application of the Kalman filter is probably the one discussed by Smith et al. (1962). Furthermore, a good and indeed interesting account of the history concerning the development of the Kalman filter as an engineering tool is given by McGee and Schmidt (1985). The aerospace industry has since the 1960s made extensive use of the Kalman filter. In Chapter 1 it was mentioned that the same trend is currently appearing in the automotive industry, due to the need for more advanced driver assistance functions. Since its first application the Kalman filter has been successively applied within many different branches of science. There are by now several application oriented texts dealing with the Kalman filter, see, e.g., Bar-Shalom and Li (1993), Bar-Shalom and Fortmann (1988), Brown and Hwang (1997), Sorenson (1985).

The linear observer theory developed by Luenberger (1966, 1971) can be considered to be a deterministic version of the Kalman filter. In the linear observer theory it is postulated that the best way to construct the state estimate is to use the following structure for the estimator

$$\hat{x}_{t+1} = A_t \hat{x}_t + B_t u_t + K_t (y_t - C_t \hat{x}_t - D_t u_t). \quad (3.32)$$

It is here important to observe a subtle, but important difference between the observer theory and the Kalman filter theory. In the former the structure (3.32) of the estimator is postulated, whereas in the latter this structure is a consequence of more elaborate assumptions and calculations, see Theorem 3.1 and Corollary 3.1. These assumptions stems from the fact that we made use of a probabilistic approach² in deriving the Kalman filter, where the errors are modeled as well, not just the deterministic dynamics. Furthermore, this implies that the gain matrix K_t is optimally³ calculated in the Kalman filter, whereas in the observer K_t has to be calculated “by hand” as a compromise between speed of reconstruction and sensitivity to disturbances. From a more practical point of view one might say that this compromise has been conveniently parameterized in terms of the design variables, which serve as tuning knobs in finding the best gain matrix for a particular problem.

There are several applications where it is required to calculate k -step ahead predictions, $k > 1$. For the general case the k -step ahead prediction is given by (3.8) and if a linear, Gaussian model (3.29) is imposed it is Gaussian. It is calculated simply by iterating (3.31c) and (3.31d) k times.

In applying the Kalman filter it is important to realize that the computations are implemented with finite-precision arithmetics, which gives rise to round-off errors. This implies that the covariance matrices might end up non-symmetric and/or indefinite. The solution to the first problem is simply to propagate only half the matrix (the elements on and below, or over, the main diagonal). The solution to the second problem is to use a square-root factorization of the covariance matrix. Hence, rather than propagating the full covariance matrix, we only propagate a square-root factor. See Kailath et al. (2000) for more details regarding this topic.

3.5.2 Smoothing

The linear filtering and prediction problems were first solved by Kalman (1960) and Kalman and Bucy (1961). It was not until a few years later that the linear smoothing problem was first solved, see Rauch (1963), Rauch et al. (1965), Bryson and Frazier (1963), Mayne (1966), Fraser and Potter (1969) for several different approaches. We will in this section only be concerned with the fixed-interval smoothing problem. The reason is threefold. First, this is the most common case in applications. Second, in the smoothing application studied in this thesis we are confronted with the fixed-interval smoothing

²In Section 3.7.1 we will use a completely deterministic approach to the estimation problem and discuss the differences and similarities between a deterministic and stochastic approach in more detail.

³The word optimal is a dangerous one. It is important to always keep in mind what is meant by optimal. The estimates are optimal in the sense that they constitute the optimal solution to the posed optimization problem. Hence, it is imperative that the optimization problem is wisely formulated, otherwise the optimal solution might not be so optimal after all.

problem⁴. Third, the solutions of the fixed-lag and the fixed-point smoothing problems follow from the solution of the fixed-interval problem (Kailath et al., 2000).

The various approaches mentioned above for solving the smoothing problem all use different arguments and as a result they produce quite different algorithms. However, since the algorithms all solve the same problem they will give the same result, which in turn implies that there must exist a close relationship between the various algorithms, enabling a unified treatment. It is the fundamental *innovation process* that makes such a unifying treatment possible, this was first recognized by Kailath and Frost (1968). A more recent discussion based on the innovation process is given in Kailath et al. (2000). Some other interesting references treating the smoothing problem are the survey papers by Meditch (1973) and Kailath (1975), and the monograph by Weinert (2001). The second corollary to Theorem 3.1 will be the linear smoothing equations (commonly referred to as the Rauch-Tung-Striebel (RTS) formulas introduced by Rauch et al. (1965)) given below.

Corollary 3.2 (Linear smoother)

Consider (3.29) and assume that the initial state is distributed as $x_0 \sim \mathcal{N}(\bar{x}_0, \bar{P}_0)$. Then, the estimate for the smoothed density function is given by

$$\hat{p}(x_t|Y_N) = \mathcal{N}(x | \hat{x}_{t|N}, P_{t|N}), \quad (3.33a)$$

where

$$\hat{x}_{t|N} = \hat{x}_{t|t} + S_t(\hat{x}_{t+1|N} - \hat{x}_{t+1|t}), \quad (3.33b)$$

$$P_{t|N} = P_{t|t} + S_t(P_{t+1|N} - P_{t+1|t})S_t^T, \quad (3.33c)$$

$$S_t = P_{t|t}A_t^T P_{t+1|t}^{-1}, \quad (3.33d)$$

where $\hat{x}_{t+1|t}$, $\hat{x}_{t|t}$, $P_{t+1|t}$ and $P_{t|t}$ are given by the Kalman filter. The initial state for the smoother is provided by the Kalman filter ($\hat{x}_{N|N}$ and $P_{N|N}$).

Proof: See Kailath et al. (2000), Rauch et al. (1965). \square

In order to obtain a numerically robust implementation of the solution to the smoothing problem we have to resort to square-root factorizations. A detailed treatment of such factorizations is given by Gibson (2003).

In extending the results to the nonlinear, non-Gaussian case it is probably a good idea to start from the general and indeed rather powerful expressions provided by the probability density functions. This will be the topic of Section 4.5. More importantly, that section will also discuss how the calculations can be performed in practice and in Paper F a successful application of the nonlinear smoothing algorithm is provided.

3.6 Improved Estimation Using Change Detection

Change detection is a well established research area concerned with the problem of detecting a change in the underlying system, see, e.g., Gustafsson (2000), Basseville and

⁴In Paper F a nonlinear fixed-interval smoothing problem has to be solved. It arises as a sub-problem when the EM algorithm is employed to solve a certain class of nonlinear system identification problems.

Nikiforov (1993), Kay (1998). This change might be due to a component failure or a change in the surrounding environment. Typically, the models employed in deriving various estimates cannot cope with all situations that might arise, but different models can be derived for the different situations. In automotive target tracking applications it is common to derive the model of the tracked vehicles based on the assumption that they stay in their own lanes. This assumption is valid most of the time, but when the tracked vehicles depart from their lanes the model is no longer correct. Hence, an interesting idea is to make use of change detection ideas to detect the lane departures and use a model that describes this motion better during the lane departure. This will improve the estimates, since a more accurate model is used. The idea is illustrated in Figure 3.1, where the de-

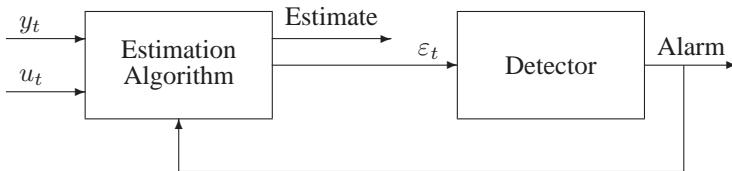


Figure 3.1: The estimation algorithm delivers residuals ε_t , which are used in the detector to decide whether or not a change has occurred. If a change is detected this information is fed back for use in the estimation algorithm.

tector informs the estimation algorithm that a change has taken place. This information is then used in the estimation algorithm by switching to the model which best describes the current situation. The change detector typically consists of a *distance measure* and a *stopping rule*, see Figure 3.2. The distance measure is used to assess whether a change

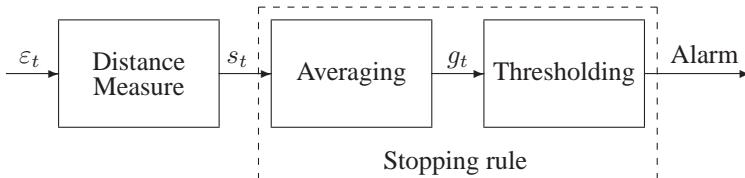


Figure 3.2: The components of the change detector are a distance measure and a stopping rule, where the latter consists of an averaging and a thresholding procedure.

has occurred or not. It is an important design variable, that should be chosen with the application in mind. Common standard choices are to use the residuals $s_t = \varepsilon_t$ or the squared residuals $s_t = \varepsilon_t^2$. The stopping rule is used to give an alarm when the *auxiliary test statistic* g_t exceeds a certain threshold. One of the most powerful tools for obtaining a good stopping rule in change detection problems is provided by the *cumulative sum (CUSUM)* algorithm, introduced by Page (1954).

Algorithm 3.2 (CUSUM)

1. $g_t = g_{t-1} + s_t - \nu.$
 2. If $g_t > h$: Alarm, $g_t = 0$ and $t_{alarm} = t.$
 3. If $g_t < 0$: $g_t = 0$ and $\hat{t}_{change} = t.$
-

The auxiliary test statistic g_t is a cumulative sum of the distance measure, compensated with a *drift term* ν . This drift term is introduced to prevent positive drifts, which otherwise will result in false alarms. Similarly, negative drifts are prevented by setting $g_t = 0$, when $g_t < 0$. The estimated change time is provided by \hat{t}_{change} . A change is considered detected when g_t exceeds a certain threshold h . A rather detailed account of the CUSUM algorithm and its application in state estimation problems is provided by Gustafsson (2000).

In Paper I we provide an application where the estimates are significantly improved by employing the change detection ideas briefly reviewed in this section. Furthermore, the importance of choosing an appropriate distance measure is illustrated.

3.7 Convex Optimization for State Estimation

The topic of this section is the use of convex optimization in solving state estimation problems. Methods based on convex optimization have been extensively used within the automatic control community in order to accommodate for the presence of constraints, using the method of *model predictive control* (MPC) (Maciejowski, 2002). However, the interest has not been that intense when it comes to the state estimation problem. Recently this has started to change, see, e.g., Goodwin (2003), Goodwin et al. (2005), Rao (2000).

In Section 3.7.1 it is illustrated that the Kalman filter is the recursive solution to a certain weighted least-squares problem. This optimization problem can then be used as a basis for extending the formulation to include constraints as well. An intuitive motivation for this approach is that if the constraints are neglected the resulting problem is reduced to the ordinary Kalman filter. This fact is utilized in Section 3.7.2 in order to illustrate how certain constraints can be taken into account in solving the estimation problem.

3.7.1 Deterministic Approach to State Estimation

This section is devoted to a purely deterministic approach to the estimation problem. In order to be able to convey the main message the discussion is limited to the linear problem. Removing the probabilistic framework previously employed will in this case simply imply that the noise terms w_t and e_t in Model 7 should be regarded as errors of unknown character. Given a set of measurements Y_t and a guess of the initial state \bar{x}_0 , the task is to determine the state x_t in such a way that it describes the obtained measurements as well as possible. That is, we are faced with a problem of curve fitting, where we want to minimize the errors $\{w_i\}_{i=0}^{t-1}$ and $\{e_i\}_{i=0}^t$, as well as the error in the initial guess, $x_0 - \bar{x}_0$. If Gauss would have been faced with this problem some 200 years ago, he would probably

have suggested us to solve the following least-squares problem

$$\begin{aligned} \min_{X_t} \quad & \|x_0 - \bar{x}_0\|_{P_0^{-1}}^2 + \sum_{i=0}^{t-1} \|w_i\|_{Q_i^{-1}}^2 + \sum_{i=0}^t \|e_i\|_{R_i^{-1}}^2 \\ \text{s.t.} \quad & x_{i+1} = A_i x_i + w_i, \quad i = 0, \dots, t-1, \\ & y_i = C_i x_i + e_i, \quad i = 0, \dots, t, \end{aligned} \quad (3.34)$$

where the weight matrices $\{Q_i\}_{i=0}^{t-1}$, $\{R_i\}_{i=0}^t$ and \bar{P}_0 are design parameters. This is a convex optimization problem, more specifically it is a *quadratic program* (QP). The theory on how to handle least-squares problems of this type is well established, see Björck (1996) and the many references therein. The estimates obtained from (3.34) are smoothed, except for the estimate of x_t , which is the filtered estimate, since we only use measurements up to and including time t .

The optimization problem stated in (3.34) can also be motivated from a probabilistic point of view by considering the problem of deriving the *maximum a posteriori* estimates for the state variables

$$\hat{X}_t = \arg \max_{X_t} p(X_t | Y_t), \quad (3.35)$$

in Model 7. The probability density function $p(X_t | Y_t)$ is proportional to $p(Y_t | X_t)p(X_t)$, where

$$p(Y_t | X_t) = \prod_{i=0}^t p(y_i | x_i) = \prod_{i=0}^t p_{e_i}(y_i - C_i x_i), \quad (3.36)$$

$$p(X_t) = p_{x_0}(x_0 - \bar{x}_0) \prod_{i=0}^{t-1} p(x_{i+1} | x_i) = p_{x_0}(x_0 - \bar{x}_0) \prod_{i=0}^{t-1} p_{w_i}(x_{i+1} - A_i x_i), \quad (3.37)$$

according to the discussion in Section 2.3.1. Putting it all together we arrive at

$$p(X_t | Y_t) = c p_{x_0}(x_0 - \bar{x}_0) \prod_{i=0}^{t-1} p_{w_i}(x_{i+1} - A_i x_i) \prod_{i=1}^t p_{e_i}(y_i - C_i x_i), \quad (3.38)$$

where $c \in \mathbf{R}^+$ derives from $p(Y_t)$. Due to the fact that the logarithmic function is strictly monotone we may consider maximizing $\log(p(X_t | Y_t))$ just as well as $p(X_t | Y_t)$. This will, together with the assumption of Gaussian noise in (3.38), give rise to the optimization problem stated in (3.34). The difference is that the weight matrices are now given by the inverse covariance matrices.

It can be proved (Rao, 2000) that the recursive solution to (3.34) is provided by the Kalman filter. The Kalman filter is in other words the recursive solution to the weighted least-squares problem (3.34). This fact will be further exploited in the subsequent section, where it is discussed how constraints can be included in the estimation problem in order to obtain better estimates. An interesting historical account of the relationship between the probabilistic formulation of the Kalman filter and the corresponding deterministic formulation is provided by Sorenson (1970).

Since we have departed from the probabilistic approach there is no way of assessing the statistical performance of the estimates. It is interesting to note that regardless of

how we formulate the estimation problem it will usually boil down to an optimization problem in a purely deterministic framework. An important difference is that the probabilistic framework provides a systematic means for choosing the design parameters, i.e., the weight matrices.

3.7.2 Constrained State Estimation

The advantage of casting the estimation problem as a convex optimization problem is that it is straightforward to add certain constraints to the problem. The theory on convex optimization is by now rather mature and there is general purpose software⁵ available for solving the resulting problems. In this way prior information about the state can be utilized, e.g., that the state is always positive or that the components of the state should sum to one, which is the case if the state is a vector of probabilities. Constraints of this type cannot be straightforwardly included in the standard Kalman filter. However, if we use the optimization problem to which the Kalman filter is the recursive solution, i.e., problem (3.34), it is straightforward to include the constraints. Here, the ideas are briefly introduced. For a more thorough treatment, see Paper D, where an example on estimating probabilities is provided. Performing state estimation using optimization techniques has previously been discussed using quadratic programs in for instance Rao et al. (2001), Rao (2000), Robertson and Lee (2002). For an introduction to constrained estimation and its connection to model predictive control (Maciejowski, 2002), see, e.g., Goodwin (2003), Goodwin et al. (2005). Both these problems are treated at a more technical level by Michalska and Mayne (1995).

The main message of convex optimization is that we should *not* differ between linear and nonlinear optimization problems, but instead between convex and non-convex problems. The class of convex problems is much larger than that covered by linear problems, and for a convex problem any local optimum is also the global optimum. A convex optimization problem is defined as

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 0, \dots, m, \\ & a_i^T x = b_i, \quad i = 0, \dots, n, \end{aligned} \tag{3.39}$$

where the functions f_0, \dots, f_m are convex and the equality constraints are linear. For a thorough introduction to convex optimization, see Boyd and Vandenberghe (2004). Motivated by the discussion in the previous section the convex optimization filtering problem can be defined according to Problem 1.

It is also worth stressing that it is straightforward to include other variables to be estimated, such as, e.g., missing data into Problem 1. Besides including them in the variables to be estimated there is probably also a need to provide some assumptions regarding how they behave, which are typically implemented as constraints.

Another type of constraints that might be interesting to add to Problem 1 are those that makes it possible to include model uncertainty. Let us assume that we are uncertain about the A -matrix in Problem 1, one way of expressing this is to say that the A -matrix should

⁵A useful and efficient software is YALMIP, developed by Löfberg (2004). It provides direct access to several of the standard numerical solvers for optimization problems, using a powerful MATLAB interface.

belong to a set of some kind. Depending on the properties of this set different optimization problems are obtained. This is in the literature referred to as *robust estimation*. For information about commonly used sets, the resulting optimization problems and how to solve them, see, e.g., El Ghaoui and Lebret (1997), Boyd and Vandenberghe (2004).

Problem 1 (Convex optimization filtering)

Assume that the densities $p_{x_0}(x_0)$, $p_{w_i}(w_i)$, and $p_{e_i}(e_i)$ are log-concave⁶. In the presence of constraints in terms of a linear dynamic Model 7, the MAP-estimate is the solution $\hat{x}_t = x_t$ to the following problem

$$\begin{aligned} \max_{X_t} \quad & \log(p_{x_0}(x_0 - \bar{x}_0)) + \sum_{i=0}^{t-1} \log(p_{w_i}(w_i)) + \sum_{i=0}^t \log(p_{e_i}(e_i)) \\ \text{s.t.} \quad & x_{i+1} = A_i x_i + w_i, \quad i = 0, \dots, t-1, \\ & y_i = C_i x_i + e_i, \quad i = 0, \dots, t. \end{aligned}$$

It is straightforward to add any convex constraints to this formulation, and the resulting problem can be solved using standard software.

The main concern with the formulation of the estimation problem given in Problem 1 is that the size of the optimization problem increases with time as more and more measurements are considered. This is unacceptable in practice and we have to find a way of bounding the number of variables. One way of doing this is to derive a recursive solution. However, when additional constraints are included this can indeed be very hard. In Zhu and Li (1999) a recursive solution is given for a special case of Problem 1 with additional constraints.

Another way of bounding the number of variables in the optimization problem is to use *moving horizon estimation* (MHE) (Maciejowski, 2002, Goodwin et al., 2005), defined in Problem 2. This is basically the same idea underpinning model predictive control, i.e., the state is estimated using a fixed size, moving window of data. A special case of this is the windowed least-squares approach discussed by Gustafsson (2000).

Problem 2 (Moving Horizon Estimation (MHE))

Assume that the densities $p_{w_i}(w_i)$ and $p_{e_i}(e_i)$ are log-concave. In the presence of constraints in terms of a linear dynamic model, the MHE-estimate is the solution $\hat{x}_t = x_t$ to the following problem

$$\begin{aligned} \max_{X_{t-L:t}} \quad & F(x_{t-L}) + \sum_{i=t-L}^{t-1} \log(p_{w_i}(w_i)) + \sum_{i=t-L+1}^t \log(p_{e_i}(e_i)) \\ \text{s.t.} \quad & x_{i+1} = A_i x_i + w_i, \quad i = t-L, \dots, t-1, \\ & y_i = C_i x_i + e_i, \quad i = t-L+1, \dots, t, \end{aligned}$$

where $F(x_{t-L})$ contains information about the past.

⁶A function $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is *log-concave* if $f(x) > 0$ for all x in the domain of f and $\log(f)$ is a concave function (Boyd and Vandenberghe, 2004).

The problem is now reduced to solving a convex optimization problem with a fixed number of variables once every time a new measurement arrives. However, it is important to understand that the approach using MHE is, in general, sub-optimal, since the influence of the past measurements is not necessarily taken care of correctly in $F(x_{t-L})$.

The formulation used in Problem 2 can probably be useful also for change detection and fault diagnosis. See Gustafsson (2001) for a similar idea using the Kalman filter over a sliding window of fixed size. In an extension to nonlinear systems a solution might be based on ideas similar to the innovation whiteness test of the filter bank approach discussed in Gustafsson (2000, Chapters 8 and 9). Furthermore, Problem 2 can be extended to the nonlinear estimation problem, by using the nonlinear Model 3 instead of the linear Model 7. The resulting problem is much harder, since it is a non-convex optimization problem. Several useful entry points into the literature on moving horizon estimation for nonlinear systems are given in Rao et al. (2001), Rao (2000).

4

Sequential Monte Carlo Methods

SEQUENTIAL Monte Carlo methods, or *particle methods*, deal with the problem of recursively estimating the probability density function $p(x_t|Y_s)$. According to the Bayesian view $p(x_t|Y_s)$ contains all statistical information available about the state variable x_t , based on the information in the measurements Y_s . This probability density function can then be used to form various state estimates according to

$$I(g(x_t)) \triangleq E\{g(x_t)|Y_s\} = \int_{\mathbf{R}^{n_x}} g(x_t) p(x_t|Y_s) dx_t. \quad (4.1)$$

The key idea underlying the sequential Monte Carlo methods is to represent the probability density function by a set of samples (also referred to as particles, hence the name particle methods) and its associated weights. The density function $p(x_t|Y_s)$ is approximated with an empirical density function,

$$p(x_t|Y_s) \approx \sum_{i=1}^M \tilde{q}_t^{(i)} \delta(x_t - x_{t|s}^{(i)}), \quad \sum_{i=1}^M \tilde{q}_t^{(i)} = 1, \quad \tilde{q}_t^{(i)} \geq 0, \quad \forall i, \quad (4.2)$$

where $\delta(\cdot)$ is the Dirac delta function and $\tilde{q}_t^{(i)}$ denotes the weight associated with particle $x_{t|s}^{(i)}$. In obtaining this approximation we have to be able to generate random numbers from complicated distributions. The approximation (4.2) can also be obtained using stochastic integration ideas, see, e.g., Geweke (1996), Bergman (1999) for such, slightly different, approaches. Even though theory states that the approximations (4.2) derived using sequential Monte Carlo methods are independent of state dimension, it matters in practice. Problems due to high dimensional state variables prevents the use of the sequential Monte Carlo methods. However, if there is a linear sub-structure available in the model equations the marginalized particle filter can be employed. It is important to note that the problem of generating random numbers from complicated distributions has previously been assessed in a *non-recursive* setting using the *Markov chain Monte Carlo methods* (MCMC).

In Section 4.1 we will make the unrealistic assumption that we can indeed generate samples from the target density. The objective of this section is to illustrate the idea and to motivate Section 4.2, which is concerned with various ideas on how to handle the fact that we cannot generate samples directly from the target density. Three different solutions to this problem are illustrated. One of these is called importance sampling resampling and this approach is used to derive the particle filter in Section 4.3. In Section 4.4 the marginalized particle filter is introduced. It can be employed when there is a linear, Gaussian sub-structure available in the model equations. The solution to the non-linear smoothing problem, using particle methods, is discussed in Section 4.5. Finally, the chapter concludes with Section 4.6 on how to obtain various estimates using (4.1).

4.1 Perfect Sampling

This section is concerned with the problem of calculating estimates (4.1) based on the assumption that we have access to M independent and identically distributed (i.i.d.) samples, $\{x^{(i)}\}_{i=1}^M$ from the target density $t(x)$. This assumption is unrealistic from a practical point of view. Nevertheless, it will allow us to illustrate the key idea underlying the sequential Monte Carlo methods. Using the samples $\{x^{(i)}\}_{i=1}^M$ an empirical estimate of the density function $t(x)$ can be formed according to

$$\hat{t}_M(x) = \sum_{i=1}^M \frac{1}{M} \delta(x - x^{(i)}). \quad (4.3)$$

Using this empirical density an estimate of $I(g(x))$ is obtained as

$$\hat{I}_M(g(x)) = \int g(x) \hat{t}_M(x) dx = \sum_{i=1}^M \frac{1}{M} g(x^{(i)}). \quad (4.4)$$

This estimate is unbiased and according to the *strong law of large numbers* we have that

$$\lim_{M \rightarrow \infty} \hat{I}_M(g(x)) \xrightarrow{\text{a.s.}} I(g(x)), \quad (4.5)$$

where $\xrightarrow{\text{a.s.}}$ denotes almost sure (a.s.) convergence (Doucet et al., 2001a). If we assume that $\sigma^2 = I(g^2(x)) - I^2(g(x)) < \infty$ the *central limit theorem* can be applied, which gives

$$\lim_{M \rightarrow \infty} \sqrt{M} \left(\hat{I}_M(g(x)) - I(g(x)) \right) \xrightarrow{d} \mathcal{N}(0, \sigma^2), \quad (4.6)$$

where \xrightarrow{d} denotes convergence in distribution (Doucet et al., 2001a). Hence, using a large number of samples, $\{x^{(i)}\}_{i=1}^M$, we can easily estimate any quantity $I(g(x))$, according to (4.4).

The assumption underlying the above discussion is that it is possible to obtain i.i.d. samples from $t(x)$. However, in practice this assumption is very seldom valid. In order to use the ideas sketched above we need to be able to generate random numbers from

complicated distributions. There has been extensive research performed regarding this problem and there are several different methods that can be used to tackle the problem.

Markov chain Monte Carlo methods are used to generate samples from probability distributions (Robert and Casella, 1999, Gilks et al., 1996). The basic idea is to generate random numbers by simulating a Markov chain, which have the target density as limit distribution. The problem with MCMC methods is that they are inherently iterative, implying that their use in solving recursive estimation problems is limited. Since we are mainly concerned with the problem of recursive estimation we have to use alternative methods. However, in the sections to come we will see that similar ideas can be used to tackle the recursive problem. In the subsequent section some of the most popular sequential Monte Carlo methods will be reviewed inspired by the framework introduced by Tanizaki (2001).

4.2 Random Number Generation

The problem under consideration in this section is to generate samples from some known probability density function, referred to as the *target density* $t(x)$. However, since we cannot generate samples from $t(x)$ directly, the idea is to employ an alternate density that is simple to draw samples from, referred to as the *sampling density* $s(x)$. The only restriction imposed on $s(x)$ is that its support should include the support of $t(x)$ ¹. When a sample $\bar{x} \sim s(x)$ is drawn the probability that it was in fact generated from the target density can be calculated. This probability can then be used to decide whether \bar{x} should be considered to be a sample from $t(x)$ or not. This probability is referred to as the *acceptance probability*, and it is typically expressed as a function of $q(\bar{x})$, defined by the following relationship,

$$t(\bar{x}) \propto q(\bar{x})s(\bar{x}). \quad (4.7)$$

Depending on the exact details of how the acceptance probability is computed different methods are obtained. The three most common methods are briefly explained below. For a more detailed explanation, see, e.g., Robert and Casella (1999), Gilks et al. (1996), Tanizaki (2001). A comparison of the three methods is provided by Liu (1996).

4.2.1 Sampling Importance Resampling

Sampling importance resampling (SIR) is an extension of an idea referred to as *importance sampling*. Hence, we will start our brief exposition on SIR by explaining the importance sampling algorithm. In discussing this algorithm the sampling density $s(x)$ is typically referred to as the *importance function*. To understand the idea behind importance sampling, note that integrals in the form (4.1) can be rewritten

$$I(g(x)) = \int_{\mathbf{R}^{n_x}} g(x) \frac{t(x)}{s(x)} s(x) dx_t. \quad (4.8)$$

¹The support of $s(x)$ includes the support of $t(x)$ if $\forall x \in \mathbf{R}^{n_x}, t(x) > 0 \Rightarrow s(x) > 0$.

Based on the discussion in Section 4.1 it is now straightforward to obtain an estimate of $I(g(x))$ by generating $M \gg 1$ samples $\{x^{(i)}\}_{i=1}^M$ from $s(x)$ and forming

$$\hat{I}_M(g(x)) = \frac{1}{M} \sum_{i=1}^M q(x^{(i)})g(x^{(i)}), \quad (4.9)$$

where

$$q(x^{(i)}) = \frac{t(x^{(i)})}{s(x^{(i)})}, \quad i = 1, \dots, M, \quad (4.10)$$

are referred to as the *importance weights*. In most state estimation applications of the importance sampling procedure the normalizing factor in the target density is unknown. This implies that the importance weights are only known up to this normalizing factor, which can be resolved by normalizing the importance weights,

$$\tilde{q}(x^{(i)}) = \frac{q(x^{(i)})}{\sum_{j=1}^M q(x^{(j)})}, \quad i = 1, \dots, M, \quad (4.11)$$

where $q(x^{(i)})$ is defined in (4.10). This normalization will for finite M introduce a bias in the estimate. However, from the strong law of large numbers the estimate is asymptotically unbiased. Details regarding this and other theoretical issues relating to the importance sampling algorithm are discussed by Geweke (1989). We have now motivated the following approximation of the target density

$$\hat{t}_M(x) = \sum_{i=1}^M \tilde{q}(x^{(i)})\delta\left(x - x^{(i)}\right). \quad (4.12)$$

The importance weights contains information about how probable it is that the corresponding sample was generated from the target density. Hence, the importance weights can be used as acceptance probabilities, which allows us to generate approximately independent samples $\{\tilde{x}^{(i)}\}_{i=1}^M$ from the target density function. The approximation $\hat{t}_M(x)$ given in (4.12) is defined using a finite number of samples $\{x^{(i)}\}_{i=1}^M$. This implies that the process of generating the samples from the target density function is limited to these samples. More specifically this is realized by *resampling* among the samples according to

$$\Pr\left(\tilde{x}^{(i)} = x^{(j)}\right) = \tilde{q}(x^{(j)}), \quad i = 1, \dots, M. \quad (4.13)$$

The SIR idea was first introduced by Rubin (1988). In Algorithm 4.1 the above discussion is summarized by describing how to approximately generate M samples from the target density.

The sampling importance resampling algorithm is closely related to the *bootstrap* procedure, introduced by Efron (1979). This relation is discussed in Smith and Gelfand (1992), where an interpretation of Algorithm 4.1 is provided in terms of a weighted bootstrap resampling procedure. It is worthwhile to note that the resampling step (4.16) is the key step when it comes to estimating density functions recursively over time. This was first realized by Gordon et al. (1993) and it will be described in detail in Section 4.3.

Algorithm 4.1 (Sampling Importance Resampling (SIR))

1. Generate M independent samples $\{x^{(i)}\}_{i=1}^M$ from $s(x)$ and compute the importance weights

$$q(x^{(i)}) = t(x^{(i)})/s(x^{(i)}), \quad i = 1, \dots, M. \quad (4.14)$$

The acceptance probabilities are now obtained by normalization

$$\tilde{q}(x^{(i)}) = q(x^{(i)}) / \sum_{j=1}^M q(x^{(j)}), \quad i = 1, \dots, M. \quad (4.15)$$

2. Generate a new set of samples $\{y^{(i)}\}_{i=1}^M$ by resampling according to

$$\Pr(\tilde{x}^{(i)} = x^{(j)}) = \tilde{q}(x^{(j)}), \quad i = 1, \dots, M. \quad (4.16)$$

4.2.2 Acceptance – Rejection Sampling

A problem inherent in the SIR algorithm is that the produced samples are only approximately distributed as the target density. This problem is not encountered by acceptance – rejection sampling, which will produce samples that are exactly distributed according to the target density. However, this algorithms suffers from several other drawbacks.

If there exists a constant $L > 0$ such that

$$t(x) \leq L s(x), \quad \forall x, \quad (4.17)$$

then Algorithm 4.2 can be used to generate M samples from the target density. A more detailed account of this algorithm is provided by Robert and Casella (1999).

Algorithm 4.2 (Acceptance – rejection sampling)

1. Generate a random number, $\tilde{x} \sim s(x)$ and compute $q(\tilde{x}) = \frac{t(\tilde{x})}{L s(\tilde{x})}$.
2. Accept \tilde{x} as a sample from $t(x)$ with probability $q(\tilde{x})$, i.e., $\Pr(x^{(i)} = \tilde{x}) = q(\tilde{x})$. If \tilde{x} is not accepted go back to step 1.
3. Repeat step 1 and 2 for $i = 1, \dots, M$.

This is the most efficient sampling method in the sense that the generated samples are mutually independent, exact draws from $t(x)$. However, as mentioned above, the algorithm suffers from some major limitations. First of all we have to find an upper bound, L , which can be quite hard. Furthermore, once this upper bound has been found it can be proved (Andrieu et al., 2001) that $\Pr(\tilde{x} \text{ accepted}) = 1/L$, which typically is a very small number. This implies that from a practical point of view the algorithm is not very useful, since on average $L \gg 1$ random numbers have to be generated in order to obtain one sample that is accepted. It is clear that we want an L which is as small as possible, motivating the choice, $L = \sup_x t(x)/s(x)$. Another, related issue is that there is no upper bound on the number of iterations required, we can only state that on average ML iterations are needed. This should be compared with the SIR algorithm, which just need M iterations. When it comes to real time applications this will of course be a major problem.

4.2.3 Metropolis – Hastings Independence Sampling

The Metropolis – Hastings algorithm is a quite general algorithm for computing estimates using the MCMC method. It was introduced by Hastings (1970), as a generalization of the algorithm proposed by Metropolis et al. (1953). An introduction to the Metropolis – Hastings algorithm is provided by Chib and Greenberg (1995). The idea of the algorithm is borrowed from acceptance – rejection sampling, in that the generated samples are either accepted or rejected. However, when a sample is rejected the current value is used as a sample from the target density. The Metropolis – Hastings *independence* sampling algorithm, which is a special case of the Metropolis – Hastings algorithm, is given in Algorithm 4.3. For a more detailed account of MCMC methods in relation to sequential Monte Carlo methods, see, e.g., Andrieu et al. (2001), Bergman (1999).

Algorithm 4.3 (Metropolis – Hastings independence sampling)

1. Initialize with $x^{(-L)} = \bar{x}$ and set $i = -L + 1$.
2. Generate $\tilde{x} \sim s(z)$ and compute the acceptance probability

$$q = \min \left(\frac{t(\tilde{x})s(x^{(i-1)})}{t(x^{(i-1)})s(\tilde{x})}, 1 \right) \quad (4.18)$$

3. Set $x^{(i)} = \tilde{x}$ with probability q . Otherwise set $x^{(i)} = x^{(i-1)}$.
4. Repeat step 2 and 3 for $i = -L + 2, \dots, M$.

The initial L samples belongs to the *burn-in* phase of the algorithm and they are automatically rejected. The reason is that the simulation has to reach its stationary phase before the samples can be considered to originate from the stationary, i.e., the target, distribution. A rather detailed analysis of Algorithm 4.3 is provided by Liu (1996).

4.3 Particle Filter

Let us consider the filtering problem, where the target density is given by the filtering density, $t(x_t) = p(x_t|Y_t)$. In order to use the idea outlined in the previous section it is necessary to choose an appropriate sampling density $s(x_t)$ and a corresponding acceptance probability. This is in fact quite simple, since from Bayes' theorem and the Markov property we have

$$p(x_t|Y_t) = p(x_t|y_t, Y_{t-1}) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \propto p(y_t|x_t)p(x_t|Y_{t-1}), \quad (4.19)$$

which suggests the following choices

$$\underbrace{p(x_t|Y_t)}_{t(x_t)} \propto \underbrace{p(y_t|x_t)}_{q(x_t)} \underbrace{p(x_t|Y_{t-1})}_{s(x_t)}. \quad (4.20)$$

The resemblance with (4.7) is obvious. Hence, we can employ the algorithms discussed in Section 4.2 to obtain samples from the target density. This provides a rather general framework for discussing particle filtering algorithms. The particle filter is typically

derived completely within an importance sampling framework, see, e.g., Doucet et al. (2000a), Liu and Chen (1998), Arulampalam et al. (2002), Schön (2003) for derivations of this kind. However, it is interesting, at least from a conceptual point of view, to note that we could just as well have used acceptance – rejection sampling, Metropolis – Hastings independence sampling or some other method to generate random numbers in order to obtain alternative particle filtering algorithms. The use of acceptance – rejection sampling is discussed by Bølviken et al. (2001) and Hürzeler and Künsch (1998). Based on the appealing properties of the sampling importance resampling idea we will choose to employ this principle in deriving the particle filter. This implies that the acceptance probabilities $\{\tilde{q}_t^{(i)}\}_{i=1}^M$ are calculated according to

$$\tilde{q}_t^{(i)} = \frac{q(x_{t|t-1}^{(i)})}{\sum_{j=1}^M q(x_{t|t-1}^{(j)})} = \frac{p(y_t|x_{t|t-1}^{(i)})}{\sum_{j=1}^M p(y_t|x_{t|t-1}^{(j)})}, \quad (4.21)$$

where $x_{t|t-1}^{(i)} \sim p(x_t|Y_{t-1})$. These predicted particles $\{x_{t|t-1}^{(i)}\}_{i=1}^M$ are generated from the underlying dynamic model and the filtered particles from the previous time instance $\{x_{t-1|t-1}^{(i)}\}_{i=1}^M$. The details behind this can be understood from the following calculation, which is a result of using the time update (3.13b) in Theorem 3.1.

$$\begin{aligned} s(x_t) &= p(x_t|Y_{t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|Y_{t-1}) dx_{t-1} \\ &\approx \int p(x_t|x_{t-1}) \sum_{i=1}^M \frac{1}{M} \delta(x_{t-1} - x_{t-1|t-1}^{(i)}) dx_{t-1} \\ &= \sum_{i=1}^M \frac{1}{M} \int p(x_t|x_{t-1}) \delta(x_{t-1} - x_{t-1|t-1}^{(i)}) dx_{t-1} \\ &= \sum_{i=1}^M \frac{1}{M} p(x_t|x_{t-1|t-1}^{(i)}). \end{aligned} \quad (4.22)$$

Hence, the predicted particles are obtained simply by passing the filtered particles through the system dynamics.

According to (4.21) the acceptance probabilities $\tilde{q}_t^{(i)}$ depends on the likelihood function $p(y_t|x_{t|t-1}^{(i)})$. This makes sense, since the likelihood reveals how likely the obtained measurement is, given the present state. The better a certain particle explains the received measurement, the higher the probability that this particle was in fact drawn from the true density. Following Algorithm 4.1, a new set of particles $\{x_{t|t}^{(i)}\}_{i=1}^M$ approximating $p(x_t|Y_t)$ is generated by resampling with replacement among the predicted particles, belonging to the sampling density

$$\Pr\left(x_{t|t}^{(i)} = x_{t|t-1}^{(j)}\right) = \tilde{q}_t^{(j)}, \quad i = 1, \dots, M. \quad (4.23)$$

If this procedure is recursively repeated over time the following approximation

$$p(x_t|Y_t) \approx \sum_{i=1}^M \frac{1}{M} \delta(x_t - x_{t|t}^{(i)}) \quad (4.24)$$

is obtained and we have in fact derived the *particle filter* algorithm, which is given in Algorithm 4.4. It was first introduced by Gordon et al. (1993). Later it was independently rediscovered by Kitagawa (1996) and Isard and Blake (1998). Some early ideas relating to the particle filter are given in Metropolis and Ulam (1949), Hammersley and Morton (1954), Akashi and Kumamoto (1977), Handschin and Mayne (1969), Handschin (1970).

Algorithm 4.4 (Particle filter)

1. Initialize the particles, $\{x_{0|t-1}^{(i)}\}_{i=1}^M \sim p_{x_0}(x_0)$ and set $t := 0$.
2. Measurement update: calculate importance weights $\{q_t^{(i)}\}_{i=1}^M$ according to

$$q_t^{(i)} = p(y_t | x_{t|t-1}^{(i)}), \quad i = 1, \dots, M, \quad (4.25)$$

and normalize $\tilde{q}_t^{(i)} = q_t^{(i)} / \sum_{j=1}^M q_t^{(j)}$.

3. Resampling: draw M particles, with replacement, according to

$$\Pr(x_{t|t}^{(i)} = x_{t|t-1}^{(j)}) = \tilde{q}_t^{(j)}, \quad i = 1, \dots, M. \quad (4.26)$$

4. Time update: predict new particles according to

$$x_{t+1|t}^{(i)} \sim p(x_{t+1|t} | x_{t|t}^{(i)}), \quad i = 1, \dots, M. \quad (4.27)$$

5. Set $t := t + 1$ and iterate from step 2.
-

First, the particle filter is initialized by drawing samples from the prior density function $p_{x_0}(x_0)$. In the measurement update the new measurement is used to assign a probability, represented by the normalized importance weight, to each particle. This probability is calculated using the likelihood function, which describes how likely it was to obtain the measurement given the information available in the particle. The normalized importance weights and the corresponding particles constitute an approximation of the filtering density. The resampling step will then return particles which are equally probable. The time update is just a matter of predicting new particles according to the system model. Furthermore, these predicted particles form the starting point for another iteration of the algorithm. There are several books available on the subject of particle filtering, see Doucet et al. (2001a), Ristic et al. (2004), Liu (2001).

4.3.1 Resampling Algorithms

The resampling step consists of drawing a new set of particles $\{x_{t|t}^{(i)}\}_{i=1}^M$ with replacement from the old particles $\{x_{t|t-1}^{(i)}\}_{i=1}^M$, in such a way that the probability of drawing $x_{t|t-1}^{(i)}$ is given by $\tilde{q}_t^{(i)}$ according to

$$\Pr(x_{t|t}^{(i)} = x_{t|t-1}^{(j)}) = \tilde{q}_t^{(j)}, \quad i = 1, \dots, M. \quad (4.28)$$

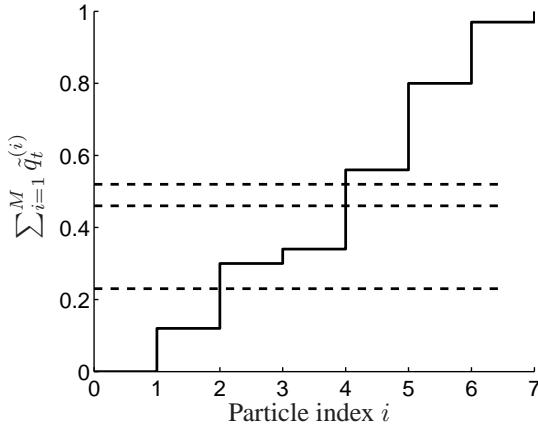


Figure 4.1: Illustrating the resampling step in the particle filter. The new set of particles is obtained by first generating M sorted uniformly distributed random numbers, three of which are shown by the dashed lines in the figure. These are then associated with a particle guided by the cumulative sum of the normalized importance weights. In the figure particle number 2 is chosen once and particle number 4 is chosen twice.

One way of achieving this is to use so called *simple random resampling*, illustrated in Figure 4.1. Here, the idea is to select the new particles by comparing an ordered set of uniformly distributed random numbers $\mathcal{U}(0, 1)$ to the cumulative sum of the normalized importance weights. The resampling step can indeed be realized according to the idea sketched in Figure 4.1, but there are more efficient algorithms available. The efficiency is here determined by the resampling quality and the computational complexity. The resampling quality is important for the overall quality of the estimate. Furthermore, a considerable amount of the total computational complexity in a particle filter implementation stems from the resampling step. This clearly motivates the search for good resampling algorithms.

There are several resampling algorithms proposed in the literature. The *simple random resampling* algorithm was explained above. For further elaboration regarding this algorithm, see Bergman (1999), Doucet et al. (2000a). Furthermore, there is *stratified sampling* (Kitagawa, 1996, Liu and Chen, 1998), *systematic sampling* (Kitagawa, 1996, Arulampalam et al., 2002) and *residual sampling* (Liu and Chen, 1998). These algorithms are discussed and analyzed in detail by Hol (2004). The result of this study is that the systematic resampling, given in Algorithm 4.5 is most appropriate. This is in accordance with the results reported by Arulampalam et al. (2002).

Despite the various embellishments of the resampling step we cannot escape the fact that it will introduce a dependence among the different particles. This is due to the fact that particles having large weights will be selected many times, since we are resampling from a discrete probability density function, rather than from a continuous. In the particle filtering literature this problem is commonly referred to as *sample impoverishment*. Theoretically this is also a problem, since this dependence makes convergence results harder

to obtain. There are several more or less *ad hoc* ideas for how to cope with this problem. One such idea is referred to as *roughening* (Gordon et al., 1993) or *jittering* (Fearnhead, 1998). The idea is to introduce an additional noise to make the particles differ more from each other. Another idea, aiming at reducing the sample impoverishment problem, is to resample from continuous approximations of the discrete probability density function. This is referred to as the *regularized particle filter* (RPF) (Musso et al., 2001).

Algorithm 4.5 (Systematic sampling)

1. Generate M ordered numbers according to

$$u_k = \frac{(k-1) + \tilde{u}}{M}, \quad \tilde{u} \sim \mathcal{U}(0, 1). \quad (4.29)$$

2. The resampled particles are obtained by producing n_i copies of particle $x^{(i)}$, where

$$n_i = \text{the number of } u_k \in \left(\sum_{s=1}^{i-1} \tilde{q}_t^{(s)}, \sum_{s=1}^i \tilde{q}_t^{(s)} \right]. \quad (4.30)$$

4.3.2 Algorithm Modifications

The particle filter given in Algorithm 4.4 is rather simple, without loosing any of the main components. In the literature there is an abundance of various alternative particle filtering algorithms. However, the underlying idea of all these algorithms is captured in Algorithm 4.4.

The essential resampling step leads to the problem of sample impoverishment, motivating the work considered with improving this part of the algorithm. An obvious idea, is to refrain from resampling at each time step. This is further discussed by Bergman (1999), where the effective sample size is used as a measure of the degeneracy of the particles. Another particle filtering algorithm devised to enhance the resampling step is the regularized particle filter mentioned above.

The importance of choosing a good importance function is stressed by several authors, see, e.g., Arulampalam et al. (2002). The importance function $p(x_{t+1}|x_t)$ used in Algorithm 4.4 has an obvious defect in the sense that the state-space is explored without direct knowledge of the measurement y_t . The idea of incorporating this information in the importance function is explored in the *auxiliary particle filter* (APF) introduced by Pitt and Shephard (1999).

The idea of approximating the probability density function with a Gaussian or a Gaussian sum was first introduced by Sorenson (1970) and Alspach and Sorenson (1972), see Section 3.4.2. This idea has recently been used within a particle filtering framework. The *Gaussian particle filter* (GPF), introduced by Kotecha and Djuric (2003a) approximates the filtering and predictive density functions with Gaussian densities. Furthermore, the *Gaussian sum particle filter* (GSPF) is similar, save the fact that the approximations are performed using a sum of Gaussian densities (Kotecha and Djuric, 2003b).

4.3.3 Implementation

The purpose of this section is to make the particle filter more accessible to those who have still not used it. Having read this section the reader will be able to implement a particle filter from scratch within five minutes. Before the implementation is given there are a few steps in Algorithm 4.4 that are probably worth commenting. In step 2 the importance weights $q_t^{(i)}$ are calculated using the likelihood function, which according to (2.14b) is given by

$$p(y_t|x_t) = p_{e_t}(y_t - h(x_t, t)). \quad (4.31)$$

Furthermore, in step 4, the task is to generate samples $x_{t+1|t}^{(i)}$ from $p(x_{t+1|t}|x_{t|t}^{(i)})$. This can be realized by first generating a sample of the process noise, $w_t^{(i)} \sim p_{w_t}(w_t)$. The predicted particles are then given by

$$x_{t+1|t}^{(i)} = f(x_{t|t}^{(i)}, t) + w_t^{(i)}. \quad (4.32)$$

We are now ready to give the MATLAB-implementation for Algorithm 4.4 using Model 3, with Gaussian noise. The resampling is implemented using Algorithm 4.5.

Code 1 (MATLAB-code for Algorithm 4.4 using Model 3)

```

function [xhat] = PF(f,h,pe,Q,P0,M,y)
n = size(P0,2);
x = sqrtm(P0)*randn(n,M); % 1. Initialize particles
for t = 1:100
    e = repmat(y(t),1,M) - h(x); % 2. Calculate weights
    q = feval(pe,e);           % The likelihood function
    q = q/sum(q);              % Normalize importance weights
    xhat(t) = sum(repmat(q,n,1).*x,2);
    ind = resampling(q);       % 3. Resampling
    x = x(:,ind);              % The new particles
    x = feval(f,x,t)+sqrtm(Q)*randn(n,M); % 4. Time update
end

function [i] = resampling(q)
qc = cumsum(q);      M=length(q);
u = ([0:M-1]+rand(1))/M;
i = zeros(1,M);      k = 1;
for j = 1:M
    while (qc(k)<u(j))
        k = k + 1;
    end
    i(j) = k;
end;

```

The three first input arguments to the `PF` function are the model equations f , h and the likelihood function pe , which are defined as `inline`-objects or m-files. The other input arguments are the covariance matrix for the state Q , initial state covariance matrix $P0$, the number of particles M and finally the measurements y . The use of Code 1 is exemplified below.

Example 4.1: State estimation using the particle filter

The purpose of this example is to show the particle filter in action in an easily accessible manner. The particle filter will be applied to estimate the states in the following system,

$$x_{t+1} = \frac{x_t}{2} + \frac{25x_t}{1+x_t^2} + 8\cos(1.2t) + w_t, \quad (4.33a)$$

$$y_t = \frac{x_t^2}{20} + e_t, \quad (4.33b)$$

where $x_0 \sim \mathcal{N}(0, 5)$, w_t and e_t are mutually independent white Gaussian noise sequences, $w_t \sim \mathcal{N}(0, 10)$ and $e_t \sim \mathcal{N}(0, 1)$. This is a discrete-time nonlinear time-varying system with additive noise, i.e., Model 3 previously defined in Section 2.3.1. This system has been analyzed in many papers, see, e.g., Gordon et al. (1993), Kitagawa (1996), Doucet (1998), Arulampalam et al. (2002).

The first step is to define the model, the parameters to use with it, and the design parameters for the particle filter. Once this is done the system is simulated and finally the measurements from this simulation are used in the particle filter to obtain the estimate of the states. The MATLAB-code for this is given below.

```

M = 1000; % Number of particles
P0 = 5; % Initial noise covariance
Q = 10; % Process noise covariance
R = 1; % Measurement noise covariance
pe = inline('1/(2*pi*1)^(1/2)*exp(-(x.^2)/(2*1))');
f = inline('x./2+25*x./(1+x.^2)+8*cos(1.2*t)','x','t');
h = inline('(x.^2)/20');

x(1) = sqrtm(P0)*randn(1); % Initial state value
y(1) = feval(h,x(1)) + sqrtm(R)*randn(1);
for t = 2:100 % Simulate the system
    x(t) = feval(f,x(t-1),t-1) + sqrtm(Q)*randn(1);
    y(t) = feval(h,x(t)) + sqrtm(R)*randn(1);
end
xTrue = x;

xhat = PF(f,h,pe,Q,P0,M,y);
plot(1:100,xhat,'b--',1:100,xTrue,'r');
xlabel('Time');

```

Executing this code gives the result shown in Figure 4.2. See Arulampalam et al. (2002) for a detailed simulation study illustrating various different particle filter algorithms.

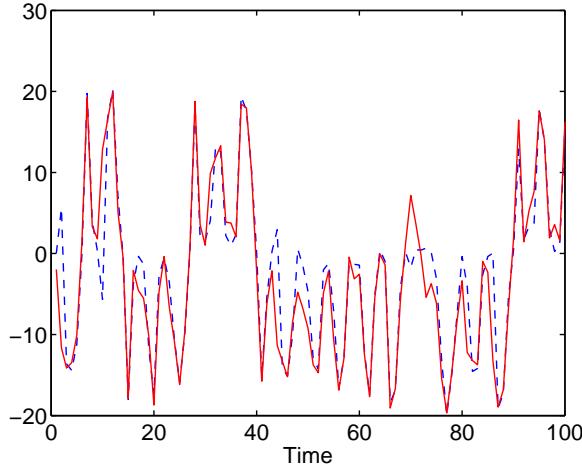


Figure 4.2: The solid line corresponds to the true state and the dashed line stems from the estimate provided by the particle filter given in Algorithm 4.4. The underlying system is given in (4.33).

The implementation given in this section is very simple, since its purpose is to as clearly as possible illustrate the particle filter. There is a toolbox available, implemented by Rosén (2005), which allows for more advanced particle filtering applications.

4.4 Marginalized Particle Filter

In mathematics, and science in general for that matter, it is often advantageous to exploit certain structures present in the problem under investigation. Sequential Monte Carlo methods are not an exception. If there is a linear, Gaussian sub-structure available in the model equations this can be used to obtain estimates with lower, or at least not larger, variance (Doucet et al., 2000a, 1999, Chen and Liu, 2000). The reason is that the corresponding linear states can be optimally estimated using the Kalman filter. Applications implying a high dimension of the state variable will effectively prevent the use of the particle filter. However, if there is a linear sub-structure available the marginalized particle filter can be used. Let us assume that there is a linear sub-structure available in the model, the state vector can then be partitioned according to

$$x_t = \begin{pmatrix} x_t^l \\ x_t^n \end{pmatrix}, \quad (4.34)$$

where x_t^l and x_t^n are used to denote the linear and the nonlinear state variables, respectively. A rather general model class containing a linear sub-structure was defined in Model 5, Section 2.3.2. The basic idea underlying the marginalized particle filter is to

split $p(x_t^l, X_t^n | Y_t)$ according to

$$p(x_t^l, X_t^n | Y_t) = p(x_t^l | X_t^n, Y_t)p(X_t^n | Y_t). \quad (4.35)$$

This allows us to use the Kalman filter to optimally estimate the probability density function for the linear variables $p(x_t^l | X_t^n, Y_t)$, if the noise is Gaussian. The probability density function for the nonlinear state variables $p(X_t^n | Y_t)$ is estimated using the particle filter. Using the state partition (4.34) it is possible to write (4.1), with $s = t$, according to

$$\begin{aligned} I(g(x_t^l, X_t)) &= \frac{1}{p(Y_t)} \int \left(\int g(x_t^l, X_t^n) p(Y_t | x_t^l, X_t^n) p(x_t^l | X_t^n) dx_t^l \right) p(X_t^n) dX_t^n \\ &= \frac{\int m(X_t^n) p(X_t^n) dX_t^n}{\int p(Y_t | X_t^n) p(X_t^n) dX_t^n}, \end{aligned} \quad (4.36)$$

where

$$m(X_t^n) \triangleq \int g(x_t^l, X_t^n) p(Y_t | x_t^l, X_t^n) p(x_t^l | X_t^n) dx_t^l. \quad (4.37)$$

Hence, we have analytically marginalized the linear state variables. This motivates the name *marginalization* for the procedure of using both the Kalman filter and the particle filter. Another name commonly used in the literature is *Rao-Blackwellization* (Casella and Robert, 1996, Doucet et al., 2000a). The idea of using a filter consisting of a Kalman filter for the linear state variables and a particle filter for the nonlinear state variables is certainly not new. It has previously been discussed in the literature, see, e.g., Doucet et al. (2000a, 2001b), Chen and Liu (2000), Nordlund (2002), Andrieu and Doucet (2002). Our contribution is the derivation of the marginalized particle filter for the rather general mixed linear/nonlinear state-space model defined as Model 5. This derivation is given in Paper A. The resulting algorithm is schematically given in Algorithm 4.6.

Algorithm 4.6 (Marginalized particle filter)

1. *Initialization:* Initialize the particles and set initial values for the linear state variables, to be used in the Kalman filter.
 2. *Particle filter measurement update:* evaluate the importance weights and normalize.
 3. *Resample with replacement.*
 4. *Particle filter time update and Kalman filter*
 - (a) *Kalman filter measurement update.*
 - (b) *Particle filter time update:* Predict new particles.
 - (c) *Kalman filter time update.*
 5. *Iterate from step 2.*
-

The only difference from the standard particle filter (Algorithm 4.1) is in step 4, where two additional steps are introduced. These two steps correspond to the efficient estimation of the linear state variables using the Kalman filter.

If the standard particle filter is used for all states the dimension of the space in which the particles live will be $n_{x_t} = \dim x_t$, whereas if the marginalized particle filter is used the corresponding dimension will be $n_{x_t^n} = \dim x_t^n$. Intuitively, since $\dim x_t^n < \dim x_t$ more particles have to be used to obtain good estimates if the particle filter is used, than if the marginalized particle filter is used. This and further issues relating to the computational complexity of the marginalized particle filter are investigated in Paper B and Karlsson et al. (2004).

The marginalized particle filter has been successfully used in several applications, for instance in automotive target tracking (Eidehall et al., 2005), automotive positioning (Svenzén, 2002), aircraft navigation (Nordlund, 2002), underwater navigation (Karlsson and Gustafsson, 2003), communications (Chen et al., 2000, Wang et al., 2002), nonlinear system identification (Paper E, Li et al., 2003, Daly et al., 2005) and audio source separation (Andrieu and Godsill, 2000). Furthermore, in Paper H the marginalized particle filter is described from a practitioners point of view, using several applications.

4.5 Particle Smoother

The aim of this section is to derive an estimate of the smoothing density $p(x_t|Y_N)$ for a fixed N and for all times, $1 \leq t \leq N$, when the underlying model is nonlinear and non-Gaussian. This is indeed a very hard problem. However, the framework discussed in Section 4.2 can be employed and will in fact provide a systematic approach to the problem. In scanning the literature it is interesting, and perhaps a bit surprising, to note that although the particle filter theory is quite well established not much work has been invested in the particle smoothing theory. Hence, this is probably a fruitful area for research during the coming years. The related Markov chain Monte Carlo methods are interesting alternatives in tackling this problem, see, e.g., Geweke and Tanizaki (1999) for some work in this direction.

4.5.1 A Particle Smoothing Algorithm

In tackling the smoothing problem the target density is chosen as (Tanizaki, 2001)

$$t(x_{t+1}, x_t) = p(x_{t+1}, x_t|Y_N). \quad (4.38)$$

Similarly to what was discussed in the Section 4.3 on particle filters, we have to find a suitable sampling density and the corresponding acceptance probabilities to solve the smoothing problem. First, note that

$$p(x_{t+1}, x_t|Y_N) = p(x_t|x_{t+1}, Y_N)p(x_{t+1}|Y_N), \quad (4.39)$$

where

$$\begin{aligned} p(x_t|x_{t+1}, Y_N) &= p(x_t|x_{t+1}, Y_t, Y_{t+1:N}) = \frac{p(Y_{t+1:N}|x_t, x_{t+1}, Y_t)p(x_t|x_{t+1}, Y_t)}{p(Y_{t+1:N}|x_{t+1}, Y_t)} \\ &= p(x_t|x_{t+1}, Y_t) = \frac{p(x_{t+1}|x_t)p(x_t|Y_t)}{p(x_{t+1}|Y_t)}. \end{aligned} \quad (4.40)$$

Inserting (4.40) into (4.39) gives

$$\underbrace{p(x_{t+1}, x_t | Y_N)}_{t(x_{t+1}, x_t)} = \underbrace{\frac{p(x_{t+1} | x_t)}{p(x_{t+1} | Y_t)}}_{q(x_{t+1}, x_t)} \underbrace{p(x_t | Y_t) p(x_{t+1} | Y_N)}_{s(x_{t+1}, x_t)} \quad (4.41)$$

At time t the sampling density can be used to generate samples. In order to find the acceptance probabilities $\{\tilde{q}^{(i)}\}_{i=1}^M$ we have to calculate

$$\tilde{q}(x_{t+1}, x_t) = \frac{p(x_{t+1} | x_t)}{p(x_{t+1} | Y_t)}, \quad (4.42)$$

where $p(x_{t+1} | x_t)$ is implied by the underlying model and $p(x_{t+1} | Y_t)$ can be approximated using the result from the particle filter,

$$\begin{aligned} p(x_{t+1} | Y_t) &= \int p(x_{t+1} | x_t) p(x_t | Y_t) dx_t = \int p(x_{t+1} | x_t) \sum_{i=1}^M \frac{1}{M} \delta(x_t - x_{t|t}^{(i)}) dx_t \\ &\approx \sum_{i=1}^M \frac{1}{M} p(x_{t+1} | x_{t|t}^{(i)}). \end{aligned} \quad (4.43)$$

The particles can now be resampled according to the acceptance probabilities $\{\tilde{q}^{(i)}\}_{i=1}^M$ in order to generate samples from $p(x_{t+1}, x_t | Y_N)$. The above discussion is summarized in Algorithm 4.7, which was first introduced by Tanizaki (2001).

Algorithm 4.7 (Particle smoother)

1. Run the particle filter (Algorithm 4.4) and store the filtered particles $\{x_{t|t}^{(i)}\}_{i=1}^M$, $t = 1, \dots, N$.
2. Initialize the smoothed particles and importance weights at time N according to $\{x_{N|N}^{(i)} = x_{N|N}^{(i)}, \tilde{q}_{N|N}^{(i)} = 1/M\}_{i=1}^M$ and set $t := t - 1$.
3. Calculate weights $\{q_{t|N}^{(i)}\}_{i=1}^M$ according to

$$q_{t|N}^{(i)} = \frac{p(x_{t+1|N}^{(i)} | x_{t|t}^{(i)})}{\sum_{j=1}^M p(x_{t+1|N}^{(j)} | x_{t|t}^{(j)})} \quad (4.44)$$

and normalize $\tilde{q}_{t|N}^{(i)} = q_{t|N}^{(i)} / \sum_{j=1}^M q_{t|N}^{(j)}$.

4. Resample the smoothed particles according to

$$\Pr \left(\left(x_{t+1|N}^{(i)}, x_{t|N}^{(i)} \right) = \left(x_{t+1|N}^{(j)}, x_{t|t}^{(j)} \right) \right) = \tilde{q}_{t|N}^{(j)}. \quad (4.45)$$

5. Set $t := t - 1$ and iterate from step 3.
-

This algorithm will be employed to handle the nonlinear smoothing problem that arises in using expectation maximization algorithm for nonlinear system identification. The idea is briefly sketched in Section 5.3.2 and the details are given in Paper F.

4.5.2 Alternative Particle Smoothing Algorithm

The algorithm just derived belongs to a set of smoothing algorithms commonly referred to as *forward-backward smoothing* algorithms. The name stems from the fact that we first perform a forward (filtering) pass to obtain an approximation of $p(x_t|Y_t)$. We then issue a backwards pass to obtain an approximation of the smoothed density $p(x_t|Y_N)$ based on the information from the forward pass and (3.13c), repeated here for convenience,

$$p(x_t|Y_N) = p(x_t|Y_t) \int_{\mathbf{R}^{n_x}} \frac{p(x_{t+1}|x_t)p(x_{t+1}|Y_N)}{p(x_{t+1}|Y_t)} dx_{t+1}. \quad (4.46)$$

This approach has also been elaborated upon by Doucet et al. (2000a), Hürzeler and Künsch (1998) and Künsch (2001).

Another set of smoothing algorithms are based on the *two-filter formula*, previously mentioned in Section 3.2. This formula describes how the marginal smoothing density can be computed by combining the output from two independent filters, according to

$$p(x_t|Y_N) \propto p(x_t|Y_{t-1})p(Y_{t:N}|x_t). \quad (4.47)$$

The details for deriving a particle smoother based on this idea is provided in Kitagawa (1996). Tanizaki's (2001) reinterpretation of the algorithm provided by Kitagawa (1996) allows us to fit this algorithm into the framework provided in Section 4.2.

The approaches discussed this far are concerned with the problem of estimating the marginal smoothing density $p(x_t|Y_N)$. We can also try to approximate the *joint* smoothing density $p(X_N|Y_N)$. An algorithm for this is proposed in Doucet et al. (2000b), Godsill et al. (2004). The idea is to factor $p(X_N|Y_N)$ according to

$$p(X_N|Y_N) = p(x_N|Y_N) \prod_{t=1}^{N-1} p(x_t|X_{t+1:N}, Y_N). \quad (4.48)$$

Using the Markov property inherent in the state-space model we have

$$\begin{aligned} p(x_t|X_{t+1:N}, Y_N) &= p(x_t|x_{t+1}, Y_t) \\ &= \frac{p(x_t|Y_t)p(x_{t+1}|x_t)}{p(x_{t+1}|Y_t)} \propto p(x_t|Y_t)p(x_{t+1}|x_t). \end{aligned} \quad (4.49)$$

Hence, it is possible to approximate $p(X_N|Y_N)$ based on the $p(x_t|Y_t)$ and $p(x_{t+1}|x_t)$. For details regarding the resulting algorithm, see Godsill et al. (2004). Some further embellishments to this algorithm are given in Fong et al. (2002), Fong and Godsill (2001), where it is discussed how marginalization can be used to derive a smoothing algorithm that exploits certain structural properties of the model.

4.6 Obtaining the Estimates

From the discussion above it is hopefully clear how to obtain estimates of probability density functions $p(x_t|Y_s)$. For instance, when $s = t$ this corresponds to the filtering density, which is approximated using the particle filter. Typically, we are interested in some

particular property of the underlying state variable, such as for instance a point estimate and its associated quality, provided by the covariance. The present section will describe how these estimates can be obtained using the approximated densities. The approach can readily be extended to other interesting features of the underlying state variable.

An minimum mean square error estimate of the mean value of the current state is obtained by inserting $g(x_t) = x_t$ in (4.1), resulting in

$$\mathbb{E}_{p(x_t|Y_s)} \{x_t\} = \int x_t p(x_t|Y_s) dx_t. \quad (4.50)$$

Using the following estimate of the probability density function,

$$\hat{p}_M(x_t|Y_s) = \sum_{i=1}^M \tilde{q}_t^{(i)} \delta \left(x_t - x_{t|s}^{(i)} \right), \quad (4.51)$$

results in

$$\hat{x}_{t|s} = \int x_t \hat{p}_M(x_t|Y_s) dx_t = \int x_t \sum_{i=1}^M \tilde{q}_t^{(i)} \delta \left(x_t - x_{t|s}^{(i)} \right) dx_t = \sum_{i=1}^M \tilde{q}_t^{(i)} x_{t|s}^{(i)}. \quad (4.52)$$

Similarly, an estimate of the covariance of $\hat{x}_{t|t}$ is obtained using

$$g(x_t) = (x_t - \hat{x}_{t|s})(x_t - \hat{x}_{t|s})^T \quad (4.53)$$

in (4.1), which after some calculations results in

$$\hat{P}_{t|s} = \sum_{i=1}^M \tilde{q}_t^{(i)} \left(x_{t|s}^{(i)} - \hat{x}_{t|s} \right) \left(x_{t|s}^{(i)} - \hat{x}_{t|s} \right)^T. \quad (4.54)$$

From the two expressions (4.52) and (4.54) it is clear how the estimates are affected by the information in the normalized importance weights $\tilde{q}_t^{(i)}$. The more likely a certain particle is, the more it influences the estimate, which is a quite reasonable fact.

5

Nonlinear System Identification

SYSTEM identification deals with the problem of estimating mathematical models of dynamic systems using measurements of the inputs to and the outputs from the system. The difference to state estimation theory is that the object to be estimated is static, which slightly changes the problem. However, both problems rely on the same theoretical basis. Similarly to the state estimation problem the system identification problem has its roots in the work of Gauss (1809) and Fisher (1912). Much of the early work was conducted within the fields of statistics, econometrics and time series analysis. It is the paper by Åström and Bohlin (1965) that is used to mark the start of *system identification* as a separate field of science. The motivation came from the field of automatic control, where new powerful model based control strategies demanded solid mathematical models of the underlying systems. An interesting historical account of the system identification problem is given by Deistler (2002). The development of the subject within the automatic control community during the past 40 years is reviewed by Gevers (2003).

In Section 5.1 an overview of the system identification problem is provided. This is followed by Section 5.2, where different methods for the model estimation process are discussed. More specifically, it is shown that the expectation maximization algorithm provides a systematic procedure for separating one hard estimation problem into two simpler problems, which is useful for system identification. Finally, in Section 5.3 the expectation maximization algorithm and particle methods are used to solve certain nonlinear system identification problems.

5.1 System Identification Problem

The system identification problem concerns estimation of static parameters present in dynamic models. This is accomplished using the information available in measured input and output signals from the underlying system. The system identification problem is

commonly split into the following sub-problems:

- **Experiment design and data collection.** This involves the selection of which variables to measure, when the measurements should be performed and how to manipulate the input signals. The objective of experiment design is to obtain data that provides as much information as possible about the parameters to be estimated.
- **Model class selection.** The problem of finding a suitable model class is the most important and probably the most difficult choice in solving an identification problem. Within the field of system identification a first, rather coarse, partition of models is constituted by *black box* and *gray box* models. In a black box model the equations and parameters do not have any physical relevance, they are simply adjusted to describe the data set as well as possible. The gray box model, on the other hand, is based on knowledge of the underlying system. Typically the model equations are known, but there are unknown parameters that have to be identified. Intuition and prior familiarity with the underlying system are very useful in choosing a suitable model class. This is true also when it comes to black box models.
- **Model estimation.** The objective is to determine the best model in the model class, using the information available in the observed data set. This is the part of the system identification problem that is considered in this thesis.
- **Model validation.** When the three steps discussed above have been performed we have derived a model. However, an important question still remains to be answered; Is the model good enough for its intended purpose? The answer to this question is obtained using model validation techniques. If the model fails the model validation some of the choices made in the previous steps have to be revised and a new model should be estimated. After a few iterations we have hopefully arrived at an acceptable model.

This is a very brief overview of the problems studied within the field of system identification, a more detailed account is provided in the monographs by Ljung (1999) and Söderström and Stoica (1989). There are also presentations solely concerned with the nonlinear system identification problem, see, e.g., Nelles (2001), Pearson (1999). The recent survey paper by Ljung (2006) provides an inventory of the nonlinear system identification problem.

5.2 Model Estimation

Depending on how the information present in the input signals $U_N = \{u_i\}_{i=1}^N$ and the output signals $Y_N = \{y_i\}_{i=1}^N$ is inferred on the parameters θ , different estimation methods are obtained. There are many different approaches to this problem and in Section 5.2.1 a very brief overview of some of the most important estimation methods is provided. In Section 5.2.2 we give a more detailed account of the expectation maximization algorithm, which is a potentially underestimated estimation method within the field of system identification.

5.2.1 Overview

Some of the most common methods used to estimate models are the *prediction error method* (Ljung, 1999, Söderström and Stoica, 1989), the *subspace method* (Van Overschee and De Moor, 1996) and the *correlation and spectral analysis methods* (Ljung, 1999). Several of these methods, and the tools to analyze their performance have their roots in, or at least strong connections to, the area of mathematical statistics.

The *maximum likelihood method*, which is a special case of the prediction error method, is quite commonly used in solving the system identification problem. It was introduced by Fisher (1912, 1922) and it is based on the rather natural idea that the parameters should be chosen in such a way that the observed measurements are *as likely as possible*. More specifically, the following optimization problem is addressed

$$\hat{\theta}(Y_N) = \arg \max_{\theta} p_{\theta}(Y_N), \quad (5.1)$$

where (recall that X_N denotes the state variables of the underlying state-space model)

$$\begin{aligned} p_{\theta}(Y_N) &= \int_{R^{Nn_x}} p_{\theta}(X_N, Y_N) dX_N = \int_{R^{Nn_x}} p_{\theta}(Y_N | X_N) p_{\theta}(X_N) dX_N \\ &= \int_{R^{Nn_x}} \prod_{t=1}^N p_{\theta}(y_t | x_t) \prod_{t=1}^N p_{\theta}(x_t | x_{t-1}) dX_N. \end{aligned} \quad (5.2)$$

Alternatively, $p_{\theta}(Y_N)$ can be written as

$$p_{\theta}(Y_N) = \prod_{t=1}^N p_{\theta}(y_t | Y_{t-1}). \quad (5.3)$$

It is often convenient to study the log-likelihood

$$L(\theta) = \log p_{\theta}(Y_N), \quad (5.4)$$

rather than the likelihood. In order to obtain an explicit optimization problem, that can be solved, we have to specify which model class we intend to use. In this thesis we only consider state-space models in the context of system identification. However, due to the need for more general models provided by differential-algebraic equations there has been some work on extending the system identification theory to handle parameter estimation in these models as well. See Gerdin (2004), Gerdin et al. (2005b) for some work in this direction.

It is interesting to see how the maximum likelihood method relates to the popular prediction error method, where the estimate is obtained as the solution to the following optimization problem

$$\hat{\theta}_N = \arg \min_{\theta} V_N(\theta, Y_N, U_N), \quad (5.5a)$$

$$V_N(\theta, Y_N, U_N) = \frac{1}{N} \sum_{t=1}^N l(\varepsilon(t, \theta)). \quad (5.5b)$$

Here, $\varepsilon(t, \theta) = y_t - \hat{y}_t$ denotes the prediction error and $l(\cdot)$ is a suitably chosen positive (norm) function. If it is chosen as

$$l(\varepsilon(t, \theta)) = -\log p_\theta(y_t | Y_{t-1}), \quad (5.6)$$

the maximum likelihood method is obtained. Hence, the prediction error method is more general than the maximum likelihood method. The use of other norms is discussed by Ljung (1999). Once the objective function has been chosen in (5.5) the optimization has to be performed. This is often a non-convex optimization problem, which typically is tackled using some gradient-based search algorithm, such as Newton's method or one of its variants¹ (Dennis and Schnabel, 1983). The iterations for the parameter estimates are typically in the following form,

$$\hat{\theta}_N^{i+1} = \hat{\theta}_N^{(i)} + \mu_N^{(i)} \left(R_N^{(i)} \right)^{-1} \left(\frac{d}{d\theta} V_N(\theta, Y_N, U_N) \right), \quad (5.7)$$

where $\mu_N^{(i)}$ is a scaling factor that denotes the step length and $R_N^{(i)}$ is a matrix that modifies the search direction. An alternative, gradient-free, solution to the maximum likelihood problem is provided by the expectation maximization algorithm, briefly introduced in the subsequent section.

5.2.2 Expectation Maximization Algorithm

The *expectation maximization* (EM) algorithm, introduced² by Dempster et al. (1977), presents an iterative approach for obtaining maximum likelihood estimates (5.1). Within the area of applied statistics it is widely recognized for its robustness. The strategy underlying the EM algorithm is to separate a difficult maximum likelihood problem into two linked problems, each of which is easier to solve than the original problem. The problems are separated using *marginalization*. It is interesting to note that this is the same underlying mechanism as in the marginalized particle filter, discussed in Section 4.4.

The key idea in the EM algorithm is to consider an extension to (5.1),

$$\hat{\theta}(X_N, Y_N) = \arg \max_{\theta} p_\theta(X_N, Y_N). \quad (5.8)$$

Here, an extra data set X_N , commonly referred to as the *incomplete data* or the *missing data*, has been introduced. Its choice is the essential design variable in devising an EM algorithm and it should be chosen in such a way that solving (5.8) is simple if X_N were known. It is worth stressing that if the missing data is chosen unwisely this might very well lead to a harder problem than what we had to begin with. The connection between (5.1) and (5.8) is provided by Bayes' theorem,

$$\log p_\theta(Y_N) = \log p_\theta(X_N, Y_N) - \log p_\theta(X_N | Y_N). \quad (5.9)$$

¹There are some special cases (FIR, ARX model structures), which give rise to a standard least-squares problem. This can of course be solved explicitly, without using an iterative approach.

²The EM algorithm was discovered independently by different researchers, see, e.g., Baum et al. (1970). However, it was Dempster et al. (1977) who provided the first systematic treatment of the ideas and introduced the name *Expectation Maximization* algorithm.

The problem separation is now obtained by marginalizing (5.9) w.r.t. the missing data. Note that θ' is used to denote the result from the previous iteration of the algorithm. Since the left-hand side of (5.9) is independent of X_N it is unaffected by the marginalization. More specifically, the marginalization is carried into effect by integrating (5.9) over $p_{\theta=\theta'}(X_N|Y_N)$. Note that $p_\theta(X_N|Y_N)$ denotes a family of density functions, parameterized by θ , whereas $p_{\theta=\theta'}(X_N|Y_N)$ denotes a specific member of this family, the one obtained using $\theta = \theta'$.

$$\begin{aligned} L(\theta) &= \log p_\theta(Y_N) = \int \log p_\theta(X_N, Y_N) p_{\theta=\theta'}(X_N|Y_N) dX_N \\ &\quad - \int \log p_\theta(X_N|Y_N) p_{\theta=\theta'}(X_N|Y_N) dX_N \\ &= \underbrace{\mathbb{E}_{\theta'} \{ \log p_\theta(X_N, Y_N) | Y_N \}}_{\mathcal{Q}(\theta, \theta')} - \underbrace{\mathbb{E}_{\theta'} \{ \log p_\theta(X_N|Y_N) | Y_N \}}_{\mathcal{V}(\theta, \theta')}, \end{aligned} \quad (5.10)$$

where $\mathbb{E}_{\theta'} \{ \cdot | Y_N \}$ denotes the expected value w.r.t. $p_{\theta=\theta'}(X_N|Y_N)$. If the log-likelihood function L is evaluated at two consecutive parameter values θ and θ' the difference can be written as

$$L(\theta) - L(\theta') = (\mathcal{Q}(\theta, \theta') - \mathcal{Q}(\theta', \theta')) + (\mathcal{V}(\theta', \theta') - \mathcal{V}(\theta, \theta')), \quad (5.11)$$

where we have made use of the definitions in (5.10). Consider the second term in (5.11),

$$\begin{aligned} \mathcal{V}(\theta', \theta') - \mathcal{V}(\theta, \theta') &= \int \log \frac{p_{\theta'}(X_N|Y_N)}{p_\theta(X_N|Y_N)} p_{\theta'}(X_N|Y_N) dX_N \\ &= \mathbb{E}_{p_{\theta'}(X_N|Y_N)} \left\{ -\log \frac{p_\theta(X_N|Y_N)}{p_{\theta'}(X_N|Y_N)} \right\}. \end{aligned} \quad (5.12)$$

It is interesting to note that $\mathcal{V}(\theta', \theta') - \mathcal{V}(\theta, \theta')$ is in fact the *Kullback-Leibler information*, which is commonly used as a measure of the agreement between two probability density functions (Kullback and Leibler, 1951). Since the negative logarithm is a convex function, Jensen's inequality³ can be used

$$\begin{aligned} \mathbb{E}_{p_{\theta'}(X_N|Y_N)} \left\{ -\log \frac{p_\theta(X_N|Y_N)}{p_{\theta'}(X_N|Y_N)} \right\} &\geq -\log \mathbb{E}_{p_{\theta'}(X_N|Y_N)} \left\{ \frac{p_\theta(X_N|Y_N)}{p_{\theta'}(X_N|Y_N)} \right\} \\ &= -\log \int p_\theta(X_N|Y_N) dX_N = 0, \end{aligned} \quad (5.13)$$

which effectively establishes that $\mathcal{V}(\theta', \theta') - \mathcal{V}(\theta, \theta') \geq 0$ and therefore choosing a θ that satisfies $\mathcal{Q}(\theta, \theta') \geq \mathcal{Q}(\theta', \theta')$ implies that $L(\theta) \geq L(\theta')$. That is, values of θ that increase $\mathcal{Q}(\theta, \theta')$ beyond its value at θ' also increase the underlying likelihood function of interest. This implies the expectation maximization algorithm stated in Algorithm 5.1.

To summarize, the EM algorithm provides a systematic procedure for separating one hard problem into two simpler connected problems, using marginalization. Given the

³Jensen's inequality (Durrett, 1996) states that if f is a convex function then

$$\mathbb{E}\{f(x)\} \geq f(\mathbb{E}\{x\})$$

Algorithm 5.1 (Expectation Maximization (EM))

Given an initial estimate θ_0 , iterate the following until convergence.

$$\mathbf{E:} \quad \mathcal{Q}(\theta, \theta_k) = \mathbb{E}_{\theta_k} \{ \log p_{\theta}(X_N, Y_N) | Y_N \}$$

$$\mathbf{M:} \quad \theta_{k+1} = \arg \max_{\theta} \mathcal{Q}(\theta, \theta_k)$$

many applications of the EM algorithm, within several other fields, it is surprising to see how little attention this algorithm has attracted within the areas of system identification and automatic control. A good entry point into the literature regarding various applications of the EM algorithm is Moon (1996) and the references therein. An early application, within the area of system identification, is given by Isaksson (1993). However, it is only recently that a thorough investigation of its use has been initiated. A rather detailed account of using the EM algorithm for estimating multivariable linear time-invariant state-space models is given by Gibson and Ninness (2005) and Gibson (2003). These results have been extended to bilinear system identification in Gibson et al. (2005). Furthermore, in Paper F we further extend the results to identify the parameters in the nonlinear Model 4, defined in Section 2.3.2. In an effort to make the EM algorithm available for solving system identification problems a toolbox has been developed by Ninness et al. (2005).

5.3 Approaches Based on Particle Methods

The problems addressed within the field of system identification exist in many other fields of science as well. This section is concerned with some new ideas on how to tackle a certain class of nonlinear system identification problems using particle methods and the EM algorithm. Hence, we will try to illustrate some new ideas based on methods extensively used in other communities for similar problems.

There is a recent survey paper by Andrieu et al. (2004), which provides an overview of the use of sequential Monte Carlo, or particle, methods in system identification, change detection and automatic control. The use of the expectation maximization within the field of system identification has been reviewed above. When the parameter estimation problem is investigated using particle methods we have implicitly made use of the Bayesian approach. This approach has previously been employed to handle the system identification problem, see, e.g., McGhee and Walford (1968), Kramer and Sorenson (1988), Peterka (1981, 1979).

The two ideas briefly introduced in the subsequent sections are concerned with the following class of nonlinear systems

$$\begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix} = \begin{pmatrix} f_1(x_t, u_t, t) \\ h_1(x_t, u_t, t) \end{pmatrix} \theta + \begin{pmatrix} f_2(x_t, u_t, t) \\ h_2(x_t, u_t, t) \end{pmatrix} + \begin{pmatrix} w_t \\ e_t \end{pmatrix}, \quad (5.14)$$

previously introduced as Model 4 in Section 2.3.1.

5.3.1 Marginalized Particle Filter

The strategy employed in this first approach is rather well-known. The idea is to augment the states with the parameters into a new state vector (Åström and Eykhoff, 1971, Ljung and Söderström, 1983)

$$z_t = \begin{pmatrix} x_t \\ \theta \end{pmatrix}. \quad (5.15)$$

By assuming a random walk variation for the parameters, the system identification problem can now be cast as a nonlinear state estimation problem, which opens up for possible use of all algorithms available for this problem. The resulting dynamic model is

$$x_{t+1} = f_1(x_t, u_t, t)\theta_t + f_2(x_t, u_t, t) + w_t, \quad (5.16a)$$

$$\theta_{t+1} = \theta_t + w_t^\theta, \quad (5.16b)$$

$$y_t = h_1(x_t, u_t, t)\theta + h_2(x_t, u_t, t) + e_t, \quad (5.16c)$$

which is a special case of Model 5, implying that the marginalized particle filter applies. Hence, this algorithm can be used to obtain a solution to the problem of identifying the parameters in model (5.14). The details of the approach are given in Paper E. A similar approach was independently proposed by Li et al. (2003), Andrieu and Godsill (2000) and it has also been employed by Daly et al. (2005). This idea has previously been explored by Ljung (1979), save the fact that the resulting state estimation problem was handled using the extended Kalman filter. The work by Kitagawa (1998) is also interesting in this context, where the parameters are estimated using a smoother rather than a filter.

5.3.2 Expectation Maximization and the Particle Smoother

The second approach is based on the expectation maximization algorithm, previously introduced in Section 5.2.2. Consider model (5.14), if the state variables x_t were known the problem of estimating the parameters θ would be rather simple. It is a standard linear regression problem. In agreement with previous applications of the EM algorithm for parameter estimation (Gibson and Ninnness, 2005), the missing data is defined to be the state sequence, $X_N \triangleq \{x_1, \dots, x_N\}$. When this choice has been made, the next step is the calculation of $\mathcal{Q}(\theta, \theta_k)$, defined in (5.10). This requires computation of the expected value of functions of the state x_t , conditional on Y_N . It is this calculation that constitutes the main difference between addressing the nonlinear and the linear problem using the EM algorithm. In the linear case, the expectations are calculated using a linear smoother. However, in the present context, we are faced with a nonlinear smoothing problem. This problem will be handled using the particle smoother given in Algorithm 4.7.

A detailed account of this approach is given in Paper F, where we also provide a simulation. This simulation indicates that the approach seems to be (perhaps) surprisingly robust to attraction to local minima. The mechanisms underlying this robustness are not yet fully understood and it is indeed a very interesting topic for future research.

5.3.3 Discussion

There is an important difference between the two approaches discussed above. It concerns the way in which the data is processed. The solution using the marginalized particle filter is, as the name reveals, a filtering solution, which is suitable for an on-line solution. The EM-based solution is on the other hand a smoothing solution, suitable only for the off-line situation. There is of course nothing that prevents the use of the on-line approach in addressing the off-line problem. However, it will restrict how the algorithm is allowed to access the data. The algorithm is only allowed to process the data sequentially, further implying that the data can only be accessed once. For the linear case this would not be a problem, but in the nonlinear case this poses a major limitation in the process of extracting all useful information from the measurements. The algorithm based on the EM algorithm and the particle smoother is, on the other hand, allowed to process the data as many times as is necessary, which allows the algorithm to analyze that data more adequate, with better estimates as a result. It should also be stressed that the first approach actually tackles a harder problem than the second approach, namely the on-line nonlinear system identification problem.

The interesting thing about the employment of the EM algorithm is that the need for particle methods arise naturally. This should be contrasted to the approach based on the marginalized particle filter, where the use of particle methods is more forced. It does not arise as a result of using standard parameter estimation methods, but rather as a result of considering another problem.

6

Concluding Remarks

IN this first part we have presented a framework for the research reported in this thesis. The aim has been to explain how the papers in Part II relate to each other and to the existing theory. In Section 6.1 the conclusions are given. There are many interesting ideas for future research, some of which are discussed in Section 6.2.

6.1 Conclusion

The work presented in this thesis has to a large extent dealt with state and parameters estimation problems arising from the mixed linear/nonlinear state-space model, introduced as Model 5. In Paper A it is explained how the marginalized particle filter can be used to solve the problem of estimating the state in this model. Several important special cases of the general model class are also discussed. In any practical application of the algorithm it is important to understand its computational complexity. Paper B provides a systematic analysis of the computational complexity of the marginalized particle filter, using the equivalent flop measure. The marginalized particle filter is discussed from a practitioners point of view in Paper H. This is accomplished by treating various positioning and target tracking applications. Furthermore, in Paper E it is discussed how to use the marginalized particle filter to solve certain system identification problems.

The system identification problem is also discussed in Paper F, where it is described how to estimate the parameters in a nonlinear state-space model, with affine parameter dependence. The approach is based on a maximum likelihood framework, where the resulting problem is solved using the expectation maximization algorithm and a particle smoothing method. The latter is used to calculate the nonlinear conditional expectations required by the expectation maximization algorithm.

All estimation algorithms discussed in this thesis are model based, stressing the need for a good model. In Paper C we propose an idea on how to incorporate white noise in

differential-algebraic equations, enabling the use of stochastic signal processing to solve various estimation problems. The main reason for studying models of this type is that they occur as a natural description from object-oriented modeling software. It is not uncommon that the model contains constraints. An approach, based on convex optimization, to handle this is presented in Paper D.

In Paper I a new approach for road geometry estimation, based on change detection methods, is given. The significantly improved performance is demonstrated using sensor data from authentic traffic environments. The problem of estimating the position and orientation of a camera is addressed in Paper G. The proposed approach is to support the inertial measurements using vision measurements, where the latter are incorporated in terms of feature displacements.

6.2 Future Research

The combination of the expectation maximization algorithm and the particle smoother deserves more attention. A systematic investigation of the hypothesis that the expectation maximization algorithm is robust towards getting trapped in local minima would probably yield interesting results. Gradient-based algorithms are prone to getting trapped in local minima, simply due to the fact that they are designed to search for minima. However, the expectation maximization algorithm is not gradient-based, there are other mechanisms guiding the search for the best estimate. We will try to apply the idea to problems of larger size in order to get a better understanding for its applicability.

The last observation in the previous paragraph naturally leads to the next topic for future research. It would be interesting to investigate how the various model classes introduced in Chapter 2 relate to other commonly used model classes. This would effectively provide a mapping between various model classes and appropriate estimation algorithms.

The combination of information from vision measurements with information from other sensors, such as radar and IMU is discussed in Chapter 1. The present approach is based on vision measurements, which are in fact *estimates* from computer vision systems. Hence, in effect, two estimation problems are solved sequentially. It would be interesting to investigate if a solution to the joint estimation problem can improve the quality of the estimates.

The idea of combining measurements from an IMU with vision measurements has been considered by many researchers. The approach used in this thesis is based on probabilistic ideas. However, the problem can equally well be approached using results from the nonlinear observer theory, see, e.g., Rehbinder (2001). There is probably a lot to be gained in trying to merge the ideas from these two branches of science in order to derive better algorithms for nonlinear state estimation/observation. There are for instance standard forms available in the nonlinear observer theory, which can prove to be useful in combination with, for instance, particle filter ideas. To give a concrete example of this we mention the possible use of the nonlinear transformations, discussed by Hou and Pugh (1999), to transform a nonlinear state-space model into a mixed linear/nonlinear state-space model. The state in this transformed model can then be estimated using the marginalized particle filter.

A

Appendix, Proof of Corollary 3.1

We will now set out to prove the Kalman filter simply by studying the general solution provided in Theorem 3.1 when the state is assumed to evolve according to a model based on linear transformation subject to Gaussian noise (defined in Model 7). This will be performed using the principle of induction. According to the assumptions $p(x_1|Y_0)$ is normal, $p(x_1|Y_0) = \mathcal{N}(x|\bar{x}_1, \bar{P}_1)$. Assume that $p(x_t|Y_{t-1}) = \mathcal{N}(x|\hat{x}_{t|t-1}, P_{t|t-1})$. The information in a new measurement can now be inferred on the state estimate using (3.13a),

$$p(x_t|Y_t) = \frac{1}{p(y_t|Y_{t-1})(2\pi)^{(n_x+n_y)/2} \sqrt{\det R_t \det P_{t|t-1}}} \cdot e^{-\frac{1}{2} \left((y_t - C_t x_t - D_t u_t)^T R_t^{-1} (y_t - C_t x_t - D_t u_t) + (x_t - \hat{x}_{t|t-1})^T P_{t|t-1}^{-1} (x_t - \hat{x}_{t|t-1}) \right)}, \quad (\text{A.1})$$

where (using marginalization)

$$p(y_t|Y_{t-1}) = \int_{\mathbf{R}^{n_x}} \frac{1}{(2\pi)^{(n_x+n_y)/2} \sqrt{\det R_t \det P_{t|t-1}}} \cdot e^{-\frac{1}{2} \left((y_t - C_t x_t - D_t u_t)^T R_t^{-1} (y_t - C_t x_t - D_t u_t) + (x_t - \hat{x}_{t|t-1})^T P_{t|t-1}^{-1} (x_t - \hat{x}_{t|t-1}) \right)} dx_t. \quad (\text{A.2})$$

In order to be able to carry out the integration above we have to isolate the integration variable, x_t . To accomplish this we will perform a change of variables,

$$\tilde{x}_{t|t-1} = x_t - \hat{x}_{t|t-1}, \quad (\text{A.3a})$$

$$\epsilon_t = y_t - C_t \hat{x}_{t|t-1} - D_t u_t. \quad (\text{A.3b})$$

The exponent in (A.2) can in terms of the new variable (A.3) be written as

$$\begin{aligned} \tilde{x}_{t|t-1}^T P_{t|t-1}^{-1} \tilde{x}_{t|t-1} + (\epsilon_t - C_t \tilde{x}_{t|t-1})^T R_t^{-1} (\epsilon_t - C_t \tilde{x}_{t|t-1}) = \\ \begin{pmatrix} \tilde{x}_{t|t-1} \\ \epsilon_t \end{pmatrix}^T \begin{pmatrix} P_{t|t-1}^{-1} + C_t R_t^{-1} C_t & -C_t^T R_t^{-1} \\ -R_t^{-1} C_t & R_t^{-1} \end{pmatrix} \begin{pmatrix} \tilde{x}_{t|t-1} \\ \epsilon_t \end{pmatrix}. \end{aligned} \quad (\text{A.4})$$

If we can write the center matrix in (A.4) as a block diagonal matrix we have in fact isolated the integration variable, since ϵ_t is independent of x_t . This can be accomplished using a block diagonal factorization (see Kailath et al., 2000, App. A) according to,

$$\begin{pmatrix} P_{t|t-1}^{-1} + C_t R_t^{-1} C_t & -C_t^T R_t^{-1} \\ -R_t^{-1} C_t & R_t^{-1} \end{pmatrix} = \begin{pmatrix} I & -K_t \\ 0 & I \end{pmatrix}^T \begin{pmatrix} P_{t|t}^{-1} & 0 \\ 0 & S_t^{-1} \end{pmatrix} \begin{pmatrix} I & -K_t \\ 0 & I \end{pmatrix}, \quad (\text{A.5})$$

where (note that S_t^{-1} is a Schur complement)

$$K_t = (P_{t|t-1}^{-1} + C_t^T R_t^{-1} C_t)^{-1} C_t^T R_t^{-1}, \quad (\text{A.6a})$$

$$P_{t|t}^{-1} = P_{t|t-1}^{-1} + C_t^T R_t^{-1} C_t, \quad (\text{A.6b})$$

$$S_t^{-1} = R_t^{-1} - R_t^{-1} C_t (P_{t|t-1}^{-1} + C_t^T R_t^{-1} C_t)^{-1} C_t^T R_t^{-1}. \quad (\text{A.6c})$$

The matrix inversion lemma¹ allows us to rewrite (A.6) according to

$$K_t = P_{t|t-1} C_t^T (R_t + C_t P_{t|t-1} C_t^T)^{-1}, \quad (\text{A.7a})$$

$$P_{t|t} = P_{t|t-1} - P_{t|t-1} C_t^T (R_t + C_t P_{t|t-1} C_t^T)^{-1} C_t P_{t|t-1}, \quad (\text{A.7b})$$

$$S_t = C_t P_{t|t-1} C_t^T + R_t. \quad (\text{A.7c})$$

Using the factorization (A.5) in (A.4) gives

$$\begin{aligned} \begin{pmatrix} \tilde{x}_{t|t-1} - K_t \epsilon_t \\ \epsilon_t \end{pmatrix}^T \begin{pmatrix} P_{t|t} & 0 \\ 0 & S_t^{-1} \end{pmatrix} \begin{pmatrix} \tilde{x}_{t|t-1} - K_t \epsilon_t \\ \epsilon_t \end{pmatrix} \\ = (\tilde{x}_{t|t-1} - K_t \epsilon_t)^T P_{t|t}^{-1} (\tilde{x}_{t|t-1} - K_t \epsilon_t) + \epsilon_t^T S_t^{-1} \epsilon_t. \end{aligned} \quad (\text{A.8})$$

The determinants in (A.2) can be written

$$\frac{1}{\det R_t \det P_{t|t-1}} = \det R_t^{-1} \det P_{t|t-1}^{-1} = \det \begin{pmatrix} P_{t|t-1}^{-1} & 0 \\ 0 & R_t^{-1} \end{pmatrix}. \quad (\text{A.9})$$

Since the determinant of a triangular matrix with unit diagonal equals one we can multiply (A.9) with any such matrix without changing the value of the expression. For example (A.9) can be written as

$$\det \left(\begin{pmatrix} I & -K_t \\ 0 & I \end{pmatrix}^{-T} \begin{pmatrix} I & 0 \\ -C_t & I \end{pmatrix}^T \begin{pmatrix} P_{t|t-1}^{-1} & 0 \\ 0 & R_t^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -C_t & I \end{pmatrix} \begin{pmatrix} I & -K_t \\ 0 & I \end{pmatrix}^{-1} \right), \quad (\text{A.10})$$

¹The matrix inversion lemma states that (Kailath et al., 2000)

$$(A - BCD)^{-1} = A^{-1} - A^{-1} D (C^{-1} + DA^{-1} B)^{-1} DA^{-1}$$

which allows us to use the block triangular factorization (A.5) to write the determinant as

$$\frac{1}{\det R_t \det P_{t|t-1}} = \det \begin{pmatrix} P_{t|t}^{-1} & 0 \\ 0 & S_t^{-1} \end{pmatrix} = \frac{1}{\det P_{t|t} \det S_t}. \quad (\text{A.11})$$

Inserting (A.8) and (A.11) into (A.2) we obtain

$$p(y_t | Y_{t-1}) = \frac{1}{(2\pi)^{n_y/2} \sqrt{\det S_t}} e^{-\frac{1}{2} \epsilon_t^T S_t^{-1} \epsilon_t}, \quad (\text{A.12})$$

after marginalization w.r.t. x_t . This expression can now be used in (A.1), which results in

$$p(x_t | Y_t) = \frac{1}{(2\pi)^{n_x/2} \sqrt{\det P_{t|t}}} e^{-\frac{1}{2} (x_t - \hat{x}_{t|t})^T P_{t|t}^{-1} (x_t - \hat{x}_{t|t})}, \quad (\text{A.13})$$

where

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t (y_t - C_t \hat{x}_{t|t-1} - D_t u_t). \quad (\text{A.14})$$

The time update (3.13b) can be written

$$p(x_{t+1} | Y_t) = \int_{\mathbf{R}^{n_x}} \frac{1}{(2\pi)^{n_x/2} \sqrt{\det Q_t \det P_{t|t}}} e^{-\frac{1}{2} ((x_{t+1} - A_t x_t - B_t u_t)^T Q_t^{-1} (x_{t+1} - A_t x_t - B_t u_t) + (x_t - \hat{x}_{t|t})^T P_{t|t}^{-1} (x_t - \hat{x}_{t|t}))} dx_t. \quad (\text{A.15})$$

This integration can be carried out if the integration variable, x_t , is isolated. This can be accomplished by a change of variables,

$$\tilde{x}_{t|t} = x_t - \hat{x}_{t|t}, \quad (\text{A.16a})$$

$$\tilde{x}_{t+1|t} = x_{t+1} - \hat{x}_{t+1|t}, \quad \text{where } \hat{x}_{t+1|t} = A_t \hat{x}_{t|t} + B_t u_t. \quad (\text{A.16b})$$

Using the triangular block factorization that was used in (A.5) gives the following expression for the exponent of (A.15),

$$\begin{aligned} & \tilde{x}_{t|t}^T P_{t|t}^{-1} \tilde{x}_{t|t} + (\tilde{x}_{t+1|t} - A_t \tilde{x}_{t|t})^T Q_t^{-1} (\tilde{x}_{t+1|t} - A_t \tilde{x}_{t|t}) \\ &= \begin{pmatrix} \tilde{x}_{t|t} \\ \tilde{x}_{t+1|t} \end{pmatrix}^T \begin{pmatrix} I & -L_t \\ 0 & I \end{pmatrix}^T \begin{pmatrix} M_t^{-1} & 0 \\ 0 & P_{t+1|t}^{-1} \end{pmatrix} \begin{pmatrix} I & -L_t \\ 0 & I \end{pmatrix} \begin{pmatrix} \tilde{x}_{t|t} \\ \tilde{x}_{t+1|t} \end{pmatrix}, \end{aligned} \quad (\text{A.17})$$

where

$$M_t = P_{t|t} - P_{t|t} A_t (Q_t + A_t P_{t|t} A_t^T)^{-1} A_t P_{t|t}, \quad (\text{A.18a})$$

$$P_{t+1|t} = A_t P_{t|t} A_t^T + Q_t, \quad (\text{A.18b})$$

$$L_t = P_{t|t} A_t^T (Q_t + A_t P_{t|t} A_t^T)^{-1}. \quad (\text{A.18c})$$

The integration (A.15) can now be performed, resulting in

$$p(x_{t+1} | Y_t) = \frac{1}{(2\pi)^{n_x/2} \sqrt{\det P_{t+1|t}}} e^{-\frac{1}{2} \tilde{x}_{t+1|t}^T P_{t+1|t}^{-1} \tilde{x}_{t+1|t}}. \quad (\text{A.19})$$

The expressions (A.7a), (A.7b), (A.14), (A.16b) and (A.18b) constitute the Kalman filter and hence the proof is complete. \square

Bibliography

- Akashi, H. and Kumamoto, H. (1977). Random sampling approach to state estimation in switching environments. *Automatica*, 13:429–434.
- Alspach, D. L. and Sorenson, H. W. (1972). Nonlinear Bayesian estimation using Gaussian sum approximation. *IEEE Transactions on Automatic Control*, 17:439–448.
- Anderson, B. D. O. and Moore, J. B. (1979). *Optimal Filtering*. Information and system science series. Prentice Hall, Englewood Cliffs, NJ, USA.
- Andersson, S. (2002). *Hidden Markov Models - Traffic Modeling and Subspace Methods*. PhD thesis, Centre for Mathematical Sciences, Mathematical Statistics, Lund University, Lund, Sweden.
- Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society*, 64(4):827–836.
- Andrieu, C., Doucet, A., and Fitzgerald, W. J. (2001). An introduction to Monte Carlo methods for Bayesian data analysis. In Mees, A. and Smith, R., editors, *Nonlinear Dynamics and Statistics*. Birkhäuser Verlag.
- Andrieu, C., Doucet, A., Singh, S. S., and Tadić, V. B. (2004). Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92(3):423–438.
- Andrieu, C. and Godsill, S. J. (2000). A particle filter for model based audio source separation. In *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000)*, Helsinki, Finland.
- Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188.

- Arulampalam, S. and Ristic, B. (2000). Comparison of the particle filter with range-parameterized and modified polar EKFs for angle-only tracking. In *Proceedings of SPIE, Signal and Data Processing of Small Target*, pages 288–299, Orlando, FL, USA.
- Ascher, U. M. and Petzold, L. R. (1998). *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, PA, USA.
- Åström, K. J. and Bohlin, T. (1965). Numerical identification of linear dynamic systems from normal operating records. In *Proceedings of IFAC Symposium on Self-Adaptive Systems*, Teddington, UK.
- Åström, K. J. and Eykhoff, P. (1971). System identification – a survey. *Automatica*, 7:123–162.
- Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., and Macintyre, B. (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47.
- Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385.
- Bar-Shalom, Y. and Fortmann, T. E. (1988). *Tracking and Data Association*. Mathematics in science and engineering. Academic Press, Orlando, FL, USA.
- Bar-Shalom, Y. and Li, X.-R. (1993). *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, Norwood, MA, USA.
- Basseville, M. and Nikiforov, I. V. (1993). *Detection of abrupt changes theory and applications*. Information and systems sciences series. Prentice Hall, Englewood Cliffs, NJ, USA.
- Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171.
- Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *The Philosophical Transactions*, 53:370–418.
- Bergman, N. (1999). *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Dissertations No 579, Linköping Studies in Science and Technology, SE-581 83 Linköping, Sweden.
- Bergman, N., Ljung, L., and Gustafsson, F. (1999). Terrain navigation using Bayesian statistics. *IEEE Control Systems Magazine*, 19(3):33–40.
- Billingsly, P. (1995). *Probability and Measure*. Wiley series in probability and mathematical statistics. Wiley interscience, New York, USA, third edition.
- Björck, Å. (1996). *Numerical methods for least squares problems*. SIAM, Philadelphia, PA, USA.

- Blackman, S. S. and Popoli, R. (1999). *Design and Analysis of Modern Tracking Systems*. Artech House, Inc., Norwood, MA, USA.
- Bølviken, E., Acklam, P. J., Christoffersen, N., and Størdal, J.-M. (2001). Monte Carlo filters for non-linear state estimation. *Automatica*, 37(2):177–183.
- Box, G. E. P. and Tiao, G. C. (1992). *Bayesian inference in statistical analysis*. John Wiley & Sons.
- Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge, UK.
- Brenan, K. E., Campbell, S. L., and Petzold, L. R. (1996). *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM's Classics in Applied Mathematics. SIAM, New York, USA.
- Bresler, Y. (1986). Two-filter formula for discrete-time non-linear bayesian smoothing. *International Journal of Control*, 43(2):626–641.
- Brown, R. G. and Hwang, P. Y. C. (1997). *Introduction to Random Signals and Applied Kalman Filtering*. Wiley, third edition.
- Bryson, A. E. and Frazier, M. (1963). Smoothing for linear and nonlinear dynamic systems. Tech. Rep. 63-119, Aero. Sys. Div. Wright-Patterson Air Force Base, Ohio.
- Bryson, A. E. and Ho, Y.-C. (1975). *Applied Optimal Control, optimization, estimation and control*. Taylor and Francis, second edition.
- Bucy, R. S. and Senne, K. D. (1971). Digital synthesis on nonlinear filters. *Automatica*, 7:287–298.
- Caarls, J., Jonker, P., and Persa, S. (2003). *Lecture Notes in Computer Science*, volume 2875, chapter Sensor Fusion for Augmented Reality, pages 160–176. Springer Verlag, Berlin, Germany.
- Campbell, S. L. (1990). Descriptor systems in the 90's. In *Proceedings of the 29th Conference on Decision and control*, pages 442–447, Honolulu, Hawaii, USA.
- Casella, G. and Robert, C. P. (1996). Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94.
- Chen, R. and Liu, J. S. (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society*, 62(3):493–508.
- Chen, R., Wang, X., and Liu, J. S. (2000). Adaptive joint detection in flat-fading channels via mixture Kalman filtering. *IEEE Transactions on Information Theory*, 46(6):2079–2094.
- Chib, S. and Greenberg, E. (1995). Understanding the Metropolis–Hastings algorithm. *The American Statistician*, 49(4):327–335.

- Chung, K. L. (1974). *A course in probability theory*, volume 21 of *Probability and mathematical statistics*. Academic Press, New York, USA, second edition.
- Crisan, D. (2001). *Particle Filters – A Theoretical Framework*, chapter 2, pages 17–41. Springer.
- Crisan, D. and Doucet, A. (2002). A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3):736–746.
- Dai, L. (1987). State estimation schemes for singular systems. In *Proceedings of the 10th IFAC World Congress*, volume 9, pages 211–215, Munich, Germany.
- Dai, L. (1989). Filtering and LQG problems for discrete-time stochastic singular systems. *IEEE Transactions on Automatic Control*, 34(10):1105–1108.
- Daly, M. J., Reilly, J. P., and Morelande, M. R. (2005). Rao-Blackwellised particle filtering for blind system identification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, USA.
- Danielsson, L. (2005). *State-of-the-art Document*. SEFS.
- Darouach, M., Zasadzinski, M., and Mehdi, D. (1993). State estimation of stochastic singular linear systems. *International Journal of Systems Sciences*, 24(2):345–354.
- Davison, A. J. (2003). Real-time simultaneous localisation and mapping with a single camera. In *Proceedings Ninth IEEE International Conference on Computer Vision*, volume 2, pages 1403–1410, Nice, France.
- Davison, A. J., Cid, Y. G., and Kita, N. (2004). Real-time 3D SLAM with wide-angle vision. In *Proceedings of the 5th IFAC/EUCON Symposium on Intelligent Autonomous Vehicles*, Lisaboa, Portugal.
- Deistler, M. (2002). *Stochastic Theory and Control*, volume 208/2002 of *Lecture Notes in Control and Information Sciences*, chapter System Identification and Time Series Analysis: Past, Present, and Future, pages 97–110. Springer. B. Pasik-Duncan (Ed.).
- Dellaert, F. and Thorpe, C. (1997). Robust car tracking using Kalman filtering and Bayesian templates. In *Proceedings of the SPIE conference on Intelligent Transportation Systems*, volume 3207.
- Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.
- Deng, Z. L. and Liu, Y. M. (1999). Descriptor Kalman filtering. *International Journal of Systems Science*, 30(11):1205–1212.
- Dennis, J. E. and Schnabel, R. B. (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall.

- Doucet, A. (1998). On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR.310, Signal Processing Group, Department of Engineering, University of Cambridge.
- Doucet, A., de Freitas, N., and Gordon, N., editors (2001a). *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, USA.
- Doucet, A., Godsill, S. J., and Andrieu, C. (2000a). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.
- Doucet, A., Godsill, S. J., and West, M. (2000b). Monte Carlo filtering and smoothing with application to time-varying spectral estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 701–704, Istanbul, Turkey.
- Doucet, A., Gordon, N., and Krishnamurthy, V. (1999). Particle filters for state estimation of jump Markov linear systems. Technical Report CUED/F-INFENG/TR 359, Signal Processing Group, Department of Engineering, University of Cambridge, Trupington street, CB2 1PZ Cambridge.
- Doucet, A., Gordon, N., and Krishnamurthy, V. (2001b). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624.
- Durrett, R. (1996). *Probability: Theory and Examples*. Duxbury Press, second edition.
- Edwards, A. W. F. (1992). *Likelihood*. John Hopkins University Press, Baltimore, USA, expanded edition. Previously published in 1972 by Cambridge University Press.
- Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, 7(1):1–26.
- Eidehall, A. (2004). *An Automotive Lane Guidance System*. Licentiate Thesis No 1122, Department of Electrical Engineering, Linköping University, Sweden.
- Eidehall, A. and Gustafsson, F. (2004). Combined road prediction and target tracking in collision avoidance. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 619–624, Parma, Italy.
- Eidehall, A. and Gustafsson, F. (2006). Obtaining reference road geometry parameters from recorded sensor data. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Tokyo, Japan. Submitted.
- Eidehall, A., Schön, T. B., and Gustafsson, F. (2005). The marginalized particle filter for automotive tracking applications. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 369–374, Las Vegas, USA.
- El Ghaoui, L. and Lebret, H. (1997). Robust solutions to least-squares problems with uncertain data. *SIAM Journal on matrix analysis and applications*, 18(4):1035–1064.
- Fearnhead, P. (1998). *Sequential Monte Carlo methods in filter theory*. PhD thesis, University of Oxford, Oxford, UK.

- Fisher, R. A. (1912). On an absolute criterion for fitting frequency curves. *Messenger of Mathematics*, 41:155–160.
- Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society Series A*, 222:309–368.
- Fong, W. and Godsill, S. (2001). Monte Carlo smoothing for non-linearly distorted signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3997–4000, Salt Lake City, USA.
- Fong, W., Godsill, S. J., Doucet, A., and West, M. (2002). Monte Carlo smoothing with application to audio signal enhancement. *IEEE Transactions on Signal Processing*, 50(2):438–449.
- Forssell, U. and Gustafsson, F. (2004). New system architectures and virtual sensors in vehicles. In *Reglermötet*, Gothenburg, Sweden.
- Fraser, D. C. and Potter, J. E. (1969). The optimum linear smoother as a combination of two optimum linear filters. *IEEE Transactions on Automatic Control*, AC-14:387–390.
- Fritzson, P. (2004). *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley-IEEE, New York, USA.
- Frykman, P. (2003). Applied particle filters in integrated aircraft navigation. Master's Thesis No LiTH-ISY-EX-3406, Department of Electrical Engineering, Linköping University, Sweden.
- Gantmacher, F. R. (1959). *The Theory of Matrices*, volume 2. Chelsea publishing company, New York, USA.
- Gauss, K. F. (1809). *Theoria motus corporum coelestium in sectionibus conicis sole ambientium*. Boston: Little, Brown and company, Boston, MA, USA. C. H. Davis, English translation: Theory of the Motion of the Heavenly Bodies Moving About the Sun in Conic Sections, 1857.
- Genc, Y., Riedel, S., Souvannavong, F., Akinlar, C., and Navab, N. (2002). Markerless tracking for AR: a learning-based approach. In *Proceedings of the International Symposium on Mixed and Augmented Reality*, pages 295–304, Darmstadt, Germany.
- Gerdin, M. (2004). *Parameter Estimation in Linear Descriptor Systems*. Licentiate Thesis No 1085, Department of Electrical Engineering, Linköping University, Sweden.
- Gerdin, M., Glad, T., and Ljung, L. (2005a). Well-posedness of filtering problems for stochastic linear DAE models. In *Proceedings 44th IEEE Conference on Decision and Control*, Seville, Spain.
- Gerdin, M., Schön, T. B., Glad, T., Gustafsson, F., and Ljung, L. (2005b). On parameter and state estimation for linear differential-algebraic equations. *Submitted to Automatica*.

- Gevers, M. (2003). A personal view on the development of system identification. In *Proceedings of the 13th IFAC Symposium on System Identification*, pages 773–784, Rotterdam, The Netherlands.
- Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339.
- Geweke, J. (1996). *Handbook of Computational Economics*, chapter Monte Carlo simulation and numerical integration, pages 731–800. Elsevier Science B.V., Amsterdam, The Netherlands. eds. H.M. Amman, D.A. Kendrick and J. Rust.
- Geweke, J. and Tanizaki, H. (1999). On Markov chain Monte Carlo methods for nonlinear and non-Gaussian state-space models. *Communications in Statistics, Simulation and Computation*, 28(4):867–894.
- Gibson, S. and Ninness, B. (2005). Robust maximum-likelihood estimation of multivariable dynamic systems. *Automatica*, 41(10):1667–1682.
- Gibson, S., Wills, A., and Ninness, B. (2005). Maximum-likelihood parameter estimation of bilinear systems. *IEEE Transactions on Automatic Control*, 50(10):1581–1596.
- Gibson, S. H. (2003). *Maximum Likelihood Estimation of Multivariable Dynamic Systems via the EM Algorithm*. PhD thesis, The University of Newcastle, Newcastle, NSW, Australia.
- Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall.
- Godsill, S. J., Doucet, A., and West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168.
- Goodwin, G. C. (2003). Inverse problems with constraints. In *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain. IFAC.
- Goodwin, G. C., Seron, M. M., and De Doná, J. A. (2005). *Constrained Control and Estimation An Optimisation Approach*. Communications and Control Engineering. Springer, London, UK.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113.
- Gustafsson, F. (2000). *Adaptive Filtering and Change Detection*. John Wiley & Sons, New York, USA.
- Gustafsson, F. (2001). Stochastic observability and fault diagnosis of additive changes in state space models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 2833–2836.
- Gustafsson, F. (2005). Statistical signal processing for automotive safety systems. In *IEEE Statistical Signal Processing Workshop*, Bordeaux, France.

- Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J. (2002). Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437.
- Gut, A. (1995). *An Intermediate Course in Probability*. Springer-Verlag.
- Hald, A. (1981). T. N. Thiele's contributions to statistics. *International Statistical Review*, 49(1):1–20.
- Hall, P. (2000). A Bayesian approach to map-aided vehicle positioning. Master's Thesis No LiTH-ISY-EX-3102, Department of Electrical Engineering, Linköping University, Sweden.
- Hammersley, J. M. and Morton, K. W. (1954). Poor man's Monte Carlo. *Journal of the Royal Statistical Society. Series B (Methodological)*, 16(1):23–38.
- Handschin, J. E. (1970). Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica*, 6:555–563.
- Handschin, J. E. and Mayne, D. Q. (1969). Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control*, 9:547–559.
- Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, Manchester, UK.
- Harvey, A. C. (1989). *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press, Cambridge, UK.
- Hastings, W. K. (1970). Monte Carlo simulation methods using Markov Chains and their applications. *Biometrika*, 57:97–109.
- Ho, Y. C. and Lee, R. C. K. (1964). A Bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, 9:333–338.
- Hol, J. (2004). Resampling in particle filters. Master's Thesis No LiTH-ISY-EX-ET-0283-2004, Department of Electrical Engineering, Linköping University, Sweden.
- Hol, J. D. (2005). Sensor fusion for camera pose estimation. Master's thesis, University of Twente, Enschede, The Netherlands.
- Hou, M. and Pugh, A. C. (1999). Observer with linear error dynamics for nonlinear multi-output systems. *Systems & Control Letters*, 37:1–9.
- Hürzeler, M. and Künsch, H. R. (1998). Monte Carlo approximations for general state-space models. *Journal of Computational and Graphical Statistics*, 7(2):175–193.
- Huster, A. (2003). *Relative position sensing by fusing monocular vision and inertial rate sensors*. PhD thesis, Department of Electrical Engineering, Stanford university.
- Isaksson, A. (1993). Identification of ARX-models subject to missing data. *IEEE Transactions on Automatic Control*, 38(5):813–819.

- Isard, M. and Blake, A. (1998). A smoothing filter for condensation. In *Proceedings of 5th European Conference on Computer Vision*, volume 1, pages 767–781.
- Isidori, A. (1989). *Nonlinear Control Systems*. Springer, Heidelberg, Germany, second edition.
- Jansson, J. (2005). *Collision Avoidance Theory with Applications to Automotive Collision Mitigation*. Dissertations No 950, Linköping Studies in Science and Technology, SE-581 83 Linköping, Sweden.
- Jazwinski, A. H. (1970). *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York, USA.
- Julier, S., Uhlmann, J., and Durrant-Whyte, H. F. (2000). A new method for nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482.
- Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422.
- Kailath, T. (1974). A view of three decades of linear filtering theory. *IEEE Transactions on Information Theory*, IT-20(2):146–181.
- Kailath, T. (1975). Supplement to “A survey of data smoothing”. *Automatica*, 11:109–111.
- Kailath, T. (1980). *Linear systems*. Prentice Hall, Englewood Cliffs, NJ, USA.
- Kailath, T. and Frost, P. (1968). An innovations approach to least-squares estimation part II: Linear smoothing in additive white noise. *IEEE Transactions on Automatic Control*, AC-13(6):655–660.
- Kailath, T., Sayed, A. H., and Hassibi, B. (2000). *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, NJ, USA.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45.
- Kalman, R. E. and Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Transactions of the ASME – Journal of Basic Engineering, Series 83D*, pages 95–108.
- Karlsson, R. (2005). *Particle Filtering for Positioning and Tracking Applications*. Dissertations No 924, Linköping Studies in Science and Technology, SE-581 83 Linköping, Sweden.
- Karlsson, R. and Gustafsson, F. (2003). Particle filter for underwater navigation. In *Proceedings of the Statistical Signal Processing Workshop*, pages 509–512, St. Louis, MO, USA.
- Karlsson, R. and Gustafsson, F. (2005). Recursive Bayesian estimation – bearings-only applications. *IEE Proceedings on Radar, Sonar, and Navigation. Special issue on target tracking: Algorithms and Applications*, 152(5):305–313.

- Karlsson, R., Schön, T., and Gustafsson, F. (2004). Complexity analysis of the marginalized particle filter. Technical Report LiTH-ISY-R-2611, Department of Electrical Engineering, Linköping University.
- Karlsson, R., Schön, T., and Gustafsson, F. (2005). Complexity analysis of the marginalized particle filter. *IEEE Transactions on Signal Processing*, 53(11):4408–4411.
- Kay, S. M. (1993). *Fundamentals of Statistical Signal Processing, Volume 1: Estimation Theory*. Prentice Hall Signal Processing. Prentice Hall, Upper Saddle River, NJ, USA.
- Kay, S. M. (1998). *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*. Prentice Hall Signal Processing. Prentice Hall, Upper Saddle River, NJ, USA.
- Khalil, H. K. (2002). *Nonlinear Systems*. Prentice Hall, NJ, USA, third edition.
- Kitagawa, G. (1991). A nonlinear smoothing method for time series analysis. *Statistica Sinica*, 1:371–388.
- Kitagawa, G. (1994). The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother. *Annals of the Institute of Statistical Mathematics*, 46(4):605–623.
- Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25.
- Kitagawa, G. (1998). A self-organizing state-space model. *Journal of the American Statistical Association*, 93(443):1203–1215.
- Klein, G. and Drummond, T. (2003). Robust visual tracking for non-instrumental augmented reality. In *International Symposium on Mixed Augmented Reality*, pages 113–123, Tokyo, Japan.
- Koch, R., Evers-Senne, J.-F., Frahm, J.-M., and Koeser, K. (2005). 3D reconstruction and rendering from image sequences. In *Proceedings of Workshop on Image Analysis For Multimedia Interactive Services*, Montreux, Switzerland.
- Kotecha, J. H. and Djuric, P. M. (2003a). Gaussian particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2592–2601.
- Kotecha, J. H. and Djuric, P. M. (2003b). Gaussian sum particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2602–2610.
- Kramer, S. C. and Sorenson, H. W. (1988). Bayesian parameter estimation. *IEEE Transactions on Automatic Control*, 33(2):217–222.
- Kulhavý, R. (1996). *Recursive Nonlinear Estimation A Geometric Approach*. Lecture Notes in Control and Information Sciences 216. Springer-Verlag, London.
- Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

- Künsch, H. R. (2001). State space and hidden Markov models. In Barndorff-Nielsen, O. E., Cox, D. R., and Klüppelberg, C., editors, *Complex Stochastic Systems*, Monographs on statistics and applied probability, pages 110–173. Chapman & Hall.
- Lauritzen, S. L. (1981). Time series analysis in 1880: A discussion of contributions made by T.N. Thiele. *International Statistical Review*, 49(3):319–331.
- Lepetit, V., Vaccagni, L., Thalmann, D., and Fua, P. (2003). Fully automated and stable registration for augmented reality applications. In *Proceedings of the second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 93–102, Tokyo, Japan.
- Li, P., Goodall, R., and Kadirkamanathan, V. (2003). Parameter estimation of railway vehicle dynamic model using Rao-Blackwellised particle filter. In *Proceedings of the European Control Conference*, Cambridge, UK.
- Li, X. R. and Jilkov, V. P. (2001). A survey of maneuvering target tracking-part III: Measurement models. In *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, pages 423–446, San Diego, USA.
- Li, X. R. and Jilkov, V. P. (2003). Survey of maneuvering target tracking-part I: dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364.
- Liu, J. S. (1996). Metropolized independent sampling with comparison to rejection ampling and importance sampling. *Statistics and Computing*, 6:113–119.
- Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer, New York, USA.
- Liu, J. S. and Chen, E. (1998). Sequential Monte Carlo methods for dynamics systems. *Journal of American Statistical Association*, 93:1032–1044.
- Ljung, L. (1979). Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control*, AC-24(1):36–50.
- Ljung, L. (1999). *System identification, Theory for the user*. System sciences series. Prentice Hall, Upper Saddle River, NJ, USA, second edition.
- Ljung, L. (2006). Some aspects on nonlinear system identification. In *Proceedings of the 14th IFAC Symposium on System Identification*, Newcastle, Australia. Accepted for publication.
- Ljung, L. and Söderström, T. (1983). *Theory and Practice of Recursive Identification*. The MIT Press series in Signal Processing, Optimization, and Control. The MIT Press, Cambridge, Massachusetts.
- Luenberger, D. G. (1966). Observers for multivariable systems. *IEEE Transactions on Automatic Control*, AC-11(2):190–197.
- Luenberger, D. G. (1971). An introduction to observers. *IEEE Transactions on Automatic Control*, AC-16(6):596–602.

- Löfberg, J. (2004). YALMIP: A toolbox for modeling and optimization in MATLAB. In *IEEE Conference on Computer Aided Control Systems Design*, Taipei, Taiwan.
- Maciejowski, J. M. (2002). *Predictive Control with Constraints*. Prentice Hall.
- MATRIS (2005). MATRIS project homepage. URL www.ist-matris.org, Accessed on November 30, 2005.
- Mattsson, S. E., Elmquist, H., and Otter, M. (1998). Physical system modeling with Modelica. *Control Engineering Practice*, 6:501–510.
- Mayne, D. Q. (1966). A solution to the smoothing problem for linear dynamic systems. *Automatica*, 4:73–92.
- McGee, L. A. and Schmidt, S. F. (1985). Discovery of the Kalman filter as a practical tool for aerospace and industry. Technical Report 86847, NASA, Ames Research Center.
- McGhee, R. B. and Walford, R. B. (1968). A Monte Carlo approach to the evaluation of conditional expectation parameter estimates for nonlinear dynamic systems. *IEEE Transactions on Automatic Control*, AC-13(1):29–37.
- Meditch, J. S. (1973). A survey of data smoothing for linear and nonlinear dynamic systems. *Automatica*, 9:151–162.
- Mendel, J. M. and Giesecking, D. L. (1971). Bibliography on the linear-quadratic-Gaussian problem. *IEEE Transactions on Automatic Control*, AC-16(6):847–869.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equations of state calculations by fast computing machine. *Journal of Chemical Physics*, 21(6):1087–1092.
- Metropolis, N. and Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341.
- Michalska, H. and Mayne, D. Q. (1995). Moving horizon observers and observer-based control. *IEEE Transactions on Automatic Control*, 40(6):995–1006.
- Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60.
- Musso, C., Oudjane, N., and Le Gland, F. (2001). Improving regularized particle filters. In Doucet, A., de Freitas, N., and Gordon, N., editors, *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, chapter 12, pages 247–271. Springer, New York.
- Nelles, O. (2001). *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer, Berlin, Germany.
- Nikoukhah, R., Campbell, S. L., and Delebecque, F. (1998). Kalman filtering for general discrete-time linear systems. In *Proceedings of the 37th Conference on Decision and Control*, pages 2886–2891, Tampa, FL, USA.

- Nikoukhah, R., Campbell, S. L., and Delebecque, F. (1999). Kalman filtering for general discrete-time linear systems. *IEEE Transactions on Automatic Control*, 44(10):1829–1839.
- Ninness, B., Wills, A., and Gibson, S. (2005). The university of Newcastle identification toolbox (UNIT). In *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic.
- Nordlund, P.-J. (2002). *Sequential Monte Carlo Filters and Integrated Navigation*. Licentiate Thesis No 945, Department of Electrical Engineering, Linköping University, Sweden.
- Nwagboso, C. O., editor (1993). *Automotive Sensory Systems*. Road Vehicle Automation Series. Chapman & Hall, London, UK.
- Øksendal, B. (2000). *Stochastic Differential Equations An Introduction with Applications*. Springer, fifth edition.
- Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1/2):100–115.
- Papoulis, A. (1991). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, third edition.
- Peach, N. (1995). Bearings-only tracking using a set of range-parameterized extended Kalman filters. In *IEE Proceedings of Control Theory and Applications*, volume 142, pages 73–80.
- Pearson, R. K. (1999). *Discrete-time dynamic models*. Topics in chemical engineering. Oxford University Press, New York, USA.
- Penski, C. (2000). A new numerical method for SDEs and its application in circuit simulation. *Journal of Computational and Applied Mathematics*, 115:461–470.
- Peterka, V. (1979). Bayesian system identification. In *Proceedings of the 5th IFAC Symposium on Identification and System Parameter Estimation*, pages 99–114, Darmstadt, Germany. Pergamon Press.
- Peterka, V. (1981). Bayesian system identification. *Automatica*, 17(1):41–53.
- Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599.
- Rao, C. V. (2000). *Moving Horizon Strategies for the Constrained Monitoring and Control of Nonlinear Discrete-Time Systems*. PhD thesis, University of Wisconsin Madison.
- Rao, C. V., Rawlings, J. B., and Lee, J. H. (2001). Constrained linear state estimation – a moving horizon approach. *Automatica*, pages 1619–1628.
- Rauch, H. E. (1963). Solutions to the linear smoothing problem. *IEEE Transactions on Automatic Control*, 8:371–372.

- Rauch, H. E., Tung, F., and Striebel, C. T. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA J.*, 3:1445–1450.
- Rehbinder, H. (2001). *State estimation and limited communication control for nonlinear robotic systems*. TRITA-MAT-01-OS-09, Royal Institute of Technology, Stockholm, Sweden.
- Ristic, B., Arulampalam, S., and Gordon, N. (2004). *Beyond the Kalman Filter: particle filters for tracking applications*. Artech House, London, UK.
- Robert, C. P. (2001). *The Bayesian choice*. Springer texts in statistics. Springer, New York, USA, second edition.
- Robert, C. P. and Casella, G. (1999). *Monte Carlo Statistical Methods*. Springer texts in statistics. Springer, New York.
- Robertson, D. G. and Lee, J. H. (2002). On the use of constraints in least squares estimation and control. *Automatica*, 38:1113–1123.
- Rosén, J. (2005). A framework for nonlinear filtering in MATLAB. Master's Thesis No LiTH-ISY-EX-05/3733-SE, Department of Electrical Engineering, Linköping University, Sweden.
- Rubin, D. B. (1988). Using the SIR algorithm to simulate posterior distributions. In Bernardo, J., DeGroot, M., Lindley, D., and Smith, A., editors, *Bayesian Statistics 3*, pages 395–402. Oxford University Press.
- Rugh, W. J. (1996). *Linear System Theory*. Information and system sciences series. Prentice Hall, Upper Saddle River, NJ, USA, second edition.
- Römisch, W. and Winkler, R. (2003). Stochastic DAEs in circuit simulation. In *Proceedings of the Oberwolfach Conference Mathematical Modelling, Simulation and Optimization of Integrated Electrical Circuits*.
- Schein, O. and Denk, G. (1998). Numerical solution of stochastic differential-algebraic equations with applications to transient noise simulation of microelectronic circuits. *Journal of Computational and Applied Mathematics*, 100:77–92.
- Schmidt, S. F. (1966). Application of state-space methods to navigation problems. *Advances in Control Systems*, 3:293–340.
- Schön, T. (2003). *On Computational Methods for Nonlinear Estimation*. Licentiate Thesis No 1047, Department of Electrical Engineering, Linköping University, Sweden.
- Schön, T., Gerdin, M., Glad, T., and Gustafsson, F. (2003a). A modeling and filtering framework for linear differential-algebraic equations. In *Proceedings of the 42nd Conference on Decision and Control*, Maui, Hawaii, USA.
- Schön, T. and Gustafsson, F. (2003). Particle filters for system identification of state-space models linear in either parameters or states. In *Proceedings of the 13th IFAC Symposium on System Identification*, pages 1287–1292, Rotterdam, The Netherlands. Invited paper.

- Schön, T., Gustafsson, F., and Hansson, A. (2003b). A note on state estimation as a convex optimization problem. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 61–64, Hong Kong.
- Schön, T., Gustafsson, F., and Nordlund, P.-J. (2005). Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289.
- Schön, T. B. and Gustafsson, F. (2005). Integrated navigation of cameras for augmented reality. In *Proceedings of the 16th IFAC world Congress*, Prague, Czech Republic.
- Schön, T. B., Karlsson, R., and Gustafsson, F. (2006a). The marginalized particle filter in practice. In *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, USA. Invited paper, accepted for publication.
- Schön, T. B., Wills, A., and Ninness, B. (2006b). Maximum likelihood nonlinear system estimation. In *Proceedings of the 14th IFAC Symposium on System Identification*, Newcastle, Australia. Accepted for publication.
- Schön, T. B., Eidehall, A., and Gustafsson, F. (2005). Lane departure detection for improved road geometry estimation. Technical Report LiTH-ISY-R-2714, Department of Electrical Engineering, Linköping University, Sweden. *Submitted to the IEEE Intelligent Vehicle Symposium*, Tokyo, Japan.
- Seal, H. L. (1967). Studies in the history of probability and statistics. XV: The historical development of the Gauss linear model. *Biometrika*, 54(1/2):1–24.
- Simon, G. and Berger, M. (2002). Reconstructing while registering: a novel approach for markerless augmented reality. In *Proceedings of the International Symposium on Mixed and Augmented Reality*, pages 285–293, Darmstadt, Germany.
- Smith, A. F. M. and Gelfand, A. E. (1992). Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician*, 46(2):84–88.
- Smith, G. L., Schmidt, S. F., and McGee, L. A. (1962). Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. Technical Report TR R-135, NASA.
- Söderström, T. (1994). *Discrete-time Stochastic Systems*. Series in Systems and Control Engineering. Prentice Hall, Cambridge, UK.
- Söderström, T. and Stoica, P. (1989). *System identification*. Systems and Control Engineering. Prentice Hall.
- Sorenson, H. W. (1970). Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum*, 7:63–68.
- Sorenson, H. W. (1974). On the development of practical nonlinear filters. *Information Sciences*, 7:253–270.
- Sorenson, H. W., editor (1985). *Kalman filtering: Theory and Application*. IEEE Press, New York, USA.

- Sorenson, H. W. and Alspach, D. L. (1971). Recursive Bayesian estimation using Gaussian sum. *Automatica*, 7:465–479.
- Spall, J. C., editor (1988). *Bayesian analysis of time series and dynamic models*, volume 94 of *Statistics: Textbooks and Monographs*. Marcel Dekker, inc., New York, USA.
- Streller, D., Fürstenberg, K., and Dietmayer, K. (2002). Vehicle and object models for robust tracking in traffic scenes using laser range images. In *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, pages 118–123, Singapore.
- Strobel, T., Servel, A., Coue, C., and Tatschke, T. (2005). Compendium on sensors – state-of-the-art of sensors and sensor data fusion for automotive preventive safety applications. Technical report, Preventive and Active Safety Systems.
- Svenzén, N. (2002). Real time implementation of map aided positioning using a Bayesian approach. Master's Thesis No LiTH-ISY-EX-3297-2002, Department of Electrical Engineering, Linköping University, Sweden.
- Tanizaki, H. (2001). Nonlinear and non-Gaussian state space modeling using sampling techniques. *Annals of the Institute of Statistical Mathematics*, 53(1):63–81.
- Thomas, G. A., Jin, J., Niblett, T., and Urquhart, C. (1997). A versatile camera position measurement system for virtual reality tv production. In *International Broadcasting Convention*, pages 284–289, Amsterdam, The Netherlands.
- Tiller, M. (2001). *Introduction to Physical Modeling with Modelica*. Kluwer Academic Press, Boston, MA, USA.
- Van Overschee, P. and De Moor, B. (1996). *Subspace identification for linear systems - theory, implementation, applications*. Kluwer Academic Publishers.
- Varga, A. (1992). Numerical algorithms and software tools for analysis and modelling of descriptor systems. In *Proceedings of the 2nd IFAC Workshop on System Structure and Control, Prague, Czechoslovakia*, pages 392–395.
- Wang, X., R. Chen, R., and Guo, D. (2002). Delayed-pilot sampling for mixture Kalman filter with application in fading channels. *IEEE Transactions on Signal Processing*, 50(2):241–254.
- Weinert, H. L. (2001). *Fixed interval smoothing for state space models*. Kluwer Academic Publishers, Norwell, USA.
- Welch, G. and Foxlin, E. (2002). Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer Graphics and Applications*, 22(6):24–38.
- West, M. and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models*. Springer series in statistics. Springer, New York, USA, second edition.

- Wiener, N. (1949). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Technology Press and Wiley, New York, USA. Originally published in 1942 as a classified National Defense Research Report.
- Winkler, R. (2003). Stochastic differential algebraic equations of index 1 and applications in circuit simulation. *Journal of Computational and Applied Mathematics*, pages 477–505.
- Yokokohji, Y., Sugawara, Y., and Yoshikawa, T. (2000). Accurate image overlay on video see-through HMDs using vision and accelerometers. In *Virtual Reality 2000 Conference*, pages 247–254, New Brunswick, NJ USA.
- You, S. and Neumann, U. (2001). Fusion of vision and gyro tracking for robust augmented reality registration. In *IEEE Virtual Reality*, pages 71–78, Yokohama, Japan.
- You, S., Neumann, U., and Azuma, R. (1999). Orientation tracking for outdoor augmented reality registration. *IEEE Computer Graphics and Applications*, 19(6):36–42.
- Zhu, Y. and Li, X. R. (1999). Recursive least squares with linear constraints. In *Proceedings of the 38th Conference on Decision & Control*, pages 2414–2419, Phoenix, AZ, USA.
- Zomotor, Z. and Franke, U. (1997). Sensor fusion for improved vision based lane recognition and object tracking with range-finders. In *Proceedings of IEEE Conference on Intelligent Transportation System*, pages 595–600, Boston, MA, USA.

Part II

Publications

Paper A

Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models

Authors: Thomas Schön, Fredrik Gustafsson, and Per-Johan Nordlund.

Edited version of paper originally published in *IEEE Transactions on Signal Processing*, 53(7), pages 2279–2289, July 2005.

Preliminary version published as Technical Report LiTH-ISY-R-2548, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models

Thomas B. Schön*, Fredrik Gustafsson*, and Per-Johan Nordlund†

*Department of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden.
E-mail: (schon, fredrik)@isy.liu.se.

†Department of Flight Data and Navigation,
Saab Aerospace,
SE-581 88 Linköping, Sweden.
E-mail: Per-Johan.Nordlund@saab.se.

Abstract

The particle filter offers a numerical tool to approximate the filtering density function for the state in nonlinear and non-Gaussian filtering problems. While the particle filter is fairly easy to implement and tune, its main drawback is that it is quite computer intensive, with the computational complexity increasing quickly with the state dimension. One remedy to this is to marginalize out the states appearing linearly in the dynamics. The result is that one Kalman filter is associated with each particle. The main contribution in this paper is the derivation of the details for the marginalized particle filter for a general nonlinear state-space model. Several important special cases occurring in typical signal processing applications are also discussed. The marginalized particle filter is applied to an integrated navigation system for aircraft. It is demonstrated that the complete high-dimensional system can be based on a particle filter using marginalization for all but three states. Excellent performance on real flight data is reported.

Keywords: Kalman filter, marginalization, navigation systems, nonlinear systems, particle filter, state estimation.

1 Introduction

THE nonlinear non-Gaussian filtering problem considered here consists of recursively computing the filter probability density function of the state vector in a general discrete-time state-space model, given the observed measurements. Such a general model can be formulated as

$$x_{t+1} = f(x_t, w_t), \quad (1a)$$

$$y_t = h(x_t, e_t). \quad (1b)$$

Here, y_t is the measurement at time t , x_t is the state variable, w_t is the process noise, e_t is the measurement noise, and f, h are two arbitrary nonlinear functions. The two noise densities p_{w_t} and p_{e_t} are independent and are assumed to be known.

The filter density $p(x_t|Y_t)$, where $Y_t = \{y_i\}_{i=0}^t$, is given by the following general measurement recursion:

$$p(x_t|Y_t) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})}, \quad (2a)$$

$$p(y_t|Y_{t-1}) = \int p(y_t|x_t)p(x_t|Y_{t-1}) dx_t, \quad (2b)$$

and the following time recursion:

$$p(x_{t+1}|Y_t) = \int p(x_{t+1}|x_t)p(x_t|Y_t) dx_t, \quad (2c)$$

initiated by $p(x_0|Y_{-1}) = p(x_0)$ (Jazwinski, 1970). For linear Gaussian models, the integrals can be solved analytically with a finite dimensional representation. This leads to the Kalman filter recursions, where the mean and the covariance matrix of the state are propagated (Anderson and Moore, 1979). More generally, no finite dimensional representation of the filter density exists. Thus, several numerical approximations of the integrals (2) have been proposed. A recent important contribution is to use simulation based methods from mathematical statistics, sequential Monte Carlo methods, commonly referred to as particle filters (Doucet et al., 2001a, 2000, Gordon et al., 1993).

Integrated navigation is used as a motivation and application example. Briefly, the integrated navigation system in the Swedish fighter aircraft Gripen consists of an inertial navigation system (INS), a terrain-aided positioning (TAP) system and an integration filter. This filter fuses the information from INS with the information from TAP. For a more thorough description of this system, see Nordlund (2002), Palmqvist (1997). TAP is currently based on a point-mass filter as presented in Bergman et al. (1999), where it is also demonstrated that the performance is quite good, close to the Cramér-Rao lower bound. Field tests conducted by the Swedish air force have confirmed the good precision. Alternatives based on the extended Kalman filter have been investigated by Bergman (1999), but have been shown to be inferior particularly in the transient phase (the EKF requires the gradient of the terrain profile, which is unambiguous only very locally). The point-mass filter, as described in Bergman et al. (1999), is likely to be changed to a marginalized particle filter in the future for Gripen.

TAP and INS are the primary sensors. Secondary sensors (GPS and so on) are used only when available and reliable. The current terrain-aided positioning filter has three states (horizontal position and heading), while the integrated navigation system estimates the accelerometer and gyroscope errors and some other states. The integration filter is currently based on a Kalman filter with 27 states, taking INS and TAP as primary input signals.

The Kalman filter which is used for integrated navigation requires Gaussian variables. However, TAP gives a multi-modal un-symmetric distribution in the Kalman filter measurement equation and it has to be approximated with a Gaussian distribution before being used in the Kalman filter. This results in severe performance degradation in many cases, and is a common cause for filter divergence and system reinitialization.

The appealing new strategy is to merge the two state vectors into one, and solve integrated navigation and terrain-aided positioning in one filter. This filter should include all 27 states, which effectively would prevent application of the particle filter. However, the state equation is almost linear, and only three states enter the measurement equation nonlinearly, namely horizontal position and heading. Once linearization (and the use of EKF) is absolutely ruled out, marginalization would be the only way to overcome the computational complexity. More generally, as soon as there is a linear sub-structure available in the general model (1) this can be utilized in order to obtain better estimates and possibly reduce the computational demand. The basic idea is to partition the state vector as

$$x_t = \begin{pmatrix} x_t^l \\ x_t^n \end{pmatrix}, \quad (3)$$

where x_t^l denotes the state variable with conditionally linear dynamics and x_t^n denotes the nonlinear state variable (Doucet et al., 2001b, Nordlund, 2002). Using Bayes' theorem we can then marginalize out the linear state variables from (1) and estimate them using the Kalman filter (Kalman, 1960), which is the optimal filter for this case. The nonlinear state variables are estimated using the particle filter. This technique is sometimes referred to as Rao-Blackwellization (Doucet et al., 2001b). The idea has been around for quite some time, see, e.g., Doucet et al. (2000), Casella and Robert (1996), Chen and Liu (2000), Andrieu and Doucet (2002), Doucet et al. (2001b), Liu (2001). The contribution of this article is the derivation of the details for a general nonlinear state-space model with a linear sub-structure. Models of this type are common and important in engineering applications, e.g., positioning, target tracking and collision avoidance (Gustafsson et al., 2002, Bar-Shalom and Li, 1993). The marginalized particle filter has been successfully used in several applications, for instance, in aircraft navigation (Nordlund, 2002), under-water navigation (Karlsson and Gustafsson, 2003), communications (Chen et al., 2000, Wang et al., 2002), nonlinear system identification (Li et al., 2003, Schön and Gustafsson, 2003), and audio source separation (Andrieu and Godsill, 2000).

Section 2 explains the idea behind using marginalization in conjunction with general linear/nonlinear state-space models. Three nested models are used in order to make the presentation easy to follow. Some important special cases and generalizations of the noise assumptions are discussed in Section 3. To illustrate the use of the marginalized particle filter, a synthetic example is given in Section 4. Finally, the application example is given in Section 5, and the conclusions are stated in Section 6.

2 Marginalization

The variance of the estimates obtained from the standard particle filter can be decreased by exploiting linear sub-structures in the model. The corresponding variables are marginalized out and estimated using an optimal linear filter. This is the main idea behind the marginalized particle filter. The goal of this section is to explain how the marginalized particle filter works by using three nested models. The models are nested in the sense that the first model is included in the second, which in turn is included in the third. The reason for presenting it in this fashion is to facilitate reader understanding, by incrementally extending the standard particle filter.

2.1 Standard Particle Filter

The particle filter is used to get an approximation of the filter density $p(x_t|Y_t)$ in the general model (1). This approximation can then be used to obtain an estimate of some inference function $g(\cdot)$ according to

$$I(g(x_t)) = \text{E}_{p(x_t|Y_t)}(g(x_t)) = \int g(x_t)p(x_t|Y_t) dx_t. \quad (4)$$

In the following, the particle filter, as it was introduced by Gordon et al. (1993), will be referred to as the standard particle filter. For a thorough introduction to the standard particle filter, see, e.g., Doucet et al. (2001a, 2000). The marginalized and the standard particle filter are closely related. The marginalized particle filter is given in Algorithm A.1 and neglecting steps 4(a) and 4(c) results in the standard particle filter algorithm.

Algorithm A.1 (Marginalized particle filter)

1. *Initialization:* For $i = 1, \dots, N$, initialize the particles, $x_{0|-1}^{n,(i)} \sim p_{x_0^n}(x_0^n)$ and set $\{x_{0|-1}^{l,(i)}, P_{0|-1}^{(i)}\} = \{\bar{x}_0^l, \bar{P}_0\}$.
2. *Particle filter measurement update:* For $i = 1, \dots, N$, evaluate the importance weights $q_t^{(i)} = p(y_t|X_t^{n,(i)}, Y_{t-1})$ and normalize

$$\tilde{q}_t^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^N q_t^{(j)}}.$$

3. *Resampling:* Resample N particles with replacement,

$$\Pr\left(x_{t|t}^{n,(i)} = x_{t|t-1}^{n,(j)}\right) = \tilde{q}_t^{(j)}.$$

4. *Particle filter time update and Kalman filter:*

- (a) *Kalman filter measurement update:*

Model 1: (10),

Model 2: (10),

Model 3: (22).

- (b) *Particle filter time update (prediction):* For $i = 1, \dots, N$, predict new particles,

$$x_{t+1|t}^{n,(i)} \sim p(x_{t+1|t}^n | X_t^{n,(i)}, Y_t).$$

(c) Kalman filter time update:

Model 1: (11),

Model 2: (17),

Model 3: (23).

5. Set $t := t + 1$ and iterate from step 2.

The particle filter algorithm A.1 is quite general and several improvements are available in the literature. It is quite common to introduce artificial noise in step 3 in order to counteract the degeneracy problem. Depending on the context various importance functions can be used in step 4(b). In Doucet et al. (2001a) several refinements to the particle filter algorithm are discussed.

2.2 Diagonal Model

The explanation of how the marginalized particle filter works is started by considering the following model.

Model 1 (Diagonal model)

$$x_{t+1}^n = f_t^n(x_t^n) + w_t^n, \quad (5a)$$

$$x_{t+1}^l = A_t^l(x_t^n)x_t^l + w_t^l, \quad (5b)$$

$$y_t = h_t(x_t^n) + C_t(x_t^n)x_t^l + e_t. \quad (5c)$$

The gaps in the equations above are placed there intentionally, in order to make the comparison to the general model (18) easier. The state noise is assumed white and Gaussian distributed according to

$$w_t = \begin{pmatrix} w_t^l \\ w_t^n \end{pmatrix} \sim \mathcal{N}(0, Q_t), \quad Q_t = \begin{pmatrix} Q_t^l & 0 \\ 0 & Q_t^n \end{pmatrix}. \quad (6a)$$

The measurement noise is assumed white and Gaussian distributed according to

$$e_t \sim \mathcal{N}(0, R_t). \quad (6b)$$

Furthermore, x_0^l is Gaussian,

$$x_0^l \sim \mathcal{N}(\bar{x}_0, \bar{P}_0). \quad (6c)$$

The density of x_0^n can be arbitrary, but it is assumed known. The A^l and C matrices are arbitrary.

Model 1 is called *diagonal model* due to the diagonal structure of the state equation (5a) and (5b). The aim of recursively estimating the filter density $p(x_t|Y_t)$ can be accomplished using the standard particle filter. However, conditioned on the nonlinear state variable x_t^n there is a linear sub-structure in (5), given by (5b). This fact can be used to

obtain better estimates of the linear states. Analytically marginalizing out the linear state variables from $p(x_t|Y_t)$ gives ($X_t^n = \{x_i^n\}_{i=0}^t$)

$$p(x_t^l, X_t^n | Y_t) = \underbrace{p(x_t^l | X_t^n, Y_t)}_{\text{Optimal KF}} \underbrace{p(X_t^n | Y_t)}_{\text{PF}}, \quad (7)$$

where $p(x_t^l | X_t^n, Y_t)$ is analytically tractable. It is given by the Kalman filter (KF), see Lemma A.1 below for the details. Furthermore, $p(X_t^n | Y_t)$ can be estimated using the particle filter (PF). If the same number of particles are used in the standard particle filter and the marginalized particle filter, the latter will, intuitively, provide better estimates. The reason for this is that the dimension of $p(x_t^n | Y_t)$ is smaller than the dimension of $p(x_t^l, x_t^n | Y_t)$, implying that the particles occupy a lower dimensional space. Another reason is that optimal algorithms are used in order to estimate the linear state variables. Let $\hat{I}_N^s(g(x_t))$ denote the estimate of (4) using the standard particle filter with N particles. When the marginalized particle filter is used the corresponding estimate is denoted by $\hat{I}_N^m(g(x_t))$. Under certain assumptions the following central limit theorem holds,

$$\sqrt{N} \left(\hat{I}_N^s(g(x_t)) - I(g(x_t)) \right) \xrightarrow{\text{d}} \mathcal{N}(0, \sigma_s^2), \quad N \rightarrow \infty, \quad (8a)$$

$$\sqrt{N} \left(\hat{I}_N^m(g(x_t)) - I(g(x_t)) \right) \xrightarrow{\text{d}} \mathcal{N}(0, \sigma_m^2), \quad N \rightarrow \infty, \quad (8b)$$

where $\sigma_s^2 \geq \sigma_m^2$. A formal proof of (8) is provided in Doucet et al. (2001b, 1999). For the sake of notational brevity the dependence of x_t^n in A_t , C_t , and h_t are suppressed below.

Lemma A.1

Given Model 1, the conditional probability density functions for $x_{t|t}^l$ and $x_{t+1|t}^l$ are given by

$$p(x_t^l | X_t^n, Y_t) = \mathcal{N}\left(x_t^l | \hat{x}_{t|t}^l, P_{t|t}\right), \quad (9a)$$

$$p(x_{t+1}^l | X_{t+1}^n, Y_t) = \mathcal{N}\left(x_{t+1}^l | \hat{x}_{t+1|t}^l, P_{t+1|t}\right), \quad (9b)$$

where

$$\hat{x}_{t|t}^l = \hat{x}_{t|t-1}^l + K_t \left(y_t - h_t - C_t \hat{x}_{t|t-1}^l \right), \quad (10a)$$

$$P_{t|t} = P_{t|t-1} - K_t C_t P_{t|t-1}, \quad (10b)$$

$$S_t = C_t P_{t|t-1} C_t^T + R_t, \quad (10c)$$

$$K_t = P_{t|t-1} C_t^T S_t^{-1}, \quad (10d)$$

and

$$\hat{x}_{t+1|t}^l = A_t^l \hat{x}_{t|t}^l, \quad (11a)$$

$$P_{t+1|t} = A_t^l P_{t|t} (A_t^l)^T + Q_t^l. \quad (11b)$$

The recursions are initiated with $\hat{x}_{0|-1}^l = \bar{x}_0$ and $P_{0|-1} = \bar{P}_0$.

Proof: We use straightforward application of the Kalman filter (Kalman, 1960, Kailath et al., 2000). \square

The second density $p(X_t^n|Y_t)$ in (7) will be approximated using the standard particle filter. Bayes' theorem and the Markov property inherent in the state-space model can be used to write $p(X_t^n|Y_t)$ as

$$p(X_t^n|Y_t) = \frac{p(y_t|X_t^n, Y_{t-1})p(x_t^n|X_{t-1}^n, Y_{t-1})}{p(y_t|Y_{t-1})} p(X_{t-1}^n|Y_{t-1}), \quad (12)$$

where an approximation of $p(X_{t-1}^n|Y_{t-1})$ is provided by the previous iteration of the particle filter. In order to perform the update (12) analytical expressions for $p(y_t|X_t^n, Y_{t-1})$ and $p(x_t^n|X_{t-1}^n, Y_{t-1})$ are needed. They are provided by the following lemma.

Lemma A.2

For Model 1, $p(y_t|X_t^n, Y_{t-1})$ and $p(x_{t+1}^n|X_t^n, Y_t)$ are given by

$$p(y_t|X_t^n, Y_{t-1}) = \mathcal{N}\left(y_t | h_t + C_t \hat{x}_{t|t-1}^l, C_t P_{t|t-1} C_t^T + R_t\right), \quad (13a)$$

$$p(x_{t+1}^n|X_t^n, Y_t) = \mathcal{N}\left(x_{t+1}^n | f_t^n, Q_t^n\right). \quad (13b)$$

Proof: We utilize basic facts about conditionally linear models; see, e.g., Harvey (1989), Schön (2003). \square

The linear system (5b) – (5c) can now be formed for each particle $\{x_t^{n,(i)}\}_{i=1}^N$ and the linear states can be estimated using the Kalman filter. This requires one Kalman filter associated with each particle. The overall algorithm for estimating the states in Model 1 is given in Algorithm A.1. From this algorithm, it should be clear that the only difference from the standard particle filter is that the time update (prediction) stage has been changed. In the standard particle filter, the prediction stage is given solely by step 4(b) in Algorithm A.1. Step 4(a) is referred to as the *measurement update* in the Kalman filter Kailath et al. (2000). Furthermore, the prediction of the nonlinear state variables $\hat{x}_{t+1|t}^n$ is obtained in step 4(b). According to (5a) the prediction of the nonlinear state variables does not contain any information about the linear state variables. This implies that $\hat{x}_{t+1|t}^n$ cannot be used to improve the quality of the estimates of the linear state variables. However, if Model 1 is generalized by imposing a dependence between the linear and the nonlinear state variables in (5a) the prediction of the nonlinear state variables can be used to improve the estimates of the linear state variables. In the subsequent section, it will be elaborated on how this affects the state estimation.

2.3 Triangular Model

Model 1 is now extended by including the term $A_t^n(x_t^n)x_t^l$ in the nonlinear state equation. This results in a *triangular model*, defined below.

Model 2 (Triangular model)

$$x_{t+1}^n = f_t^n(x_t^n) + A_t^n x_t^l + w_t^n, \quad (14a)$$

$$x_{t+1}^l = A_t^l(x_t^n) x_t^l + w_t^l, \quad (14b)$$

$$y_t = h_t(x_t^n) + C_t(x_t^n) x_t^l + e_t, \quad (14c)$$

with the same assumptions as in Model 1.

Now, from (14a), it is clear that $\hat{x}_{t+1|t}^n$ does indeed contain information about the linear state variables. This implies that there will be information about the linear state variable x_t^l in the prediction of the nonlinear state variable $\hat{x}_{t+1|t}^n$. To understand how this affects the derivation, it is assumed that step 4(b) in Algorithm A.1 has just been completed. This means that the predictions $\hat{x}_{t+1|t}^n$ are available, and the model can be written (the information in the measurement y_t has already been used in step 4(a))

$$x_{t+1}^l = A_t^l x_t^l + w_t^l, \quad (15a)$$

$$z_t = A_t^n x_t^l + w_t^n, \quad (15b)$$

where

$$z_t = x_{t+1}^n - f_t^n. \quad (15c)$$

It is possible to interpret z_t as a measurement and w_t^n as the corresponding measurement noise. Since (15) is a linear state-space model with Gaussian noise, the optimal state estimate is given by the Kalman filter according to

$$\hat{x}_{t|t}^{l*} = \hat{x}_{t|t}^l + L_t (z_t - A_t^n \hat{x}_{t|t}^l), \quad (16a)$$

$$P_{t|t}^* = P_{t|t} - L_t N_t L_t^T, \quad (16b)$$

$$L_t = P_{t|t}(A_t^n)^T N_t^{-1}, \quad (16c)$$

$$N_t = A_t^n P_{t|t}(A_t^n)^T + Q_t^n, \quad (16d)$$

where “*” has been used to distinguish this second measurement update from the first one. Furthermore, $\hat{x}_{t|t}^l$ and $P_{t|t}$ are given by (10a) and (10b), respectively. The final step is to merge this second measurement update with the time update to obtain the predicted states. This results in

$$\hat{x}_{t+1|t}^l = A_t^l \hat{x}_{t|t}^l + L_t (z_t - A_t^n \hat{x}_{t|t}^l), \quad (17a)$$

$$P_{t+1|t} = A_t^l P_{t|t} (A_t^l)^T + Q_t^l - L_t N_t L_t^T, \quad (17b)$$

$$L_t = A_t^l P_{t|t} (A_t^n)^T N_t^{-1}, \quad (17c)$$

$$N_t = A_t^n P_{t|t} (A_t^n)^T + Q_t^n. \quad (17d)$$

For a formal proof of this, see the Appendix. To make Algorithm A.1 valid for the more general Model 2, the time update equation in the Kalman filter (11) has to be replaced by (17).

The second measurement update is called measurement update due to the fact that the mathematical structure is the same as a measurement update in the Kalman filter. However, strictly speaking, it is not really a measurement update, since there does not exist any new measurement. It is better to think of this second update as a correction to the real measurement update using the information in the prediction of the nonlinear state variables.

2.4 General Case

In the previous two sections, the mechanisms underlying the marginalized particle filter have been illustrated. It is now time to apply the marginalized particle filter to the most general model.

Model 3 (General model)

$$x_{t+1}^n = f_t^n(x_t^n) + A_t^n(x_t^n)x_t^l + G_t^n(x_t^n)w_t^n, \quad (18a)$$

$$x_{t+1}^l = f_t^l(x_t^n) + A_t^l(x_t^n)x_t^l + G_t^l(x_t^n)w_t^l, \quad (18b)$$

$$y_t = h_t(x_t^n) + C_t(x_t^n)x_t^l + e_t, \quad (18c)$$

where the state noise is assumed white and Gaussian distributed with

$$w_t = \begin{pmatrix} w_t^l \\ w_t^n \end{pmatrix} \sim \mathcal{N}(0, Q_t), \quad Q_t = \begin{pmatrix} Q_t^l & Q_t^{ln} \\ (Q_t^{ln})^T & Q_t^n \end{pmatrix}. \quad (19a)$$

The measurement noise is assumed white and Gaussian distributed according to

$$e_t \sim \mathcal{N}(0, R_t). \quad (19b)$$

Furthermore, x_0^l is Gaussian

$$x_0^l \sim \mathcal{N}(\bar{x}_0, \bar{P}_0). \quad (19c)$$

The density of x_0^n can be arbitrary, but it is assumed known.

In certain cases, some of the assumptions can be relaxed. This will be discussed in the subsequent section. Before moving on it is worthwhile to explain how models used in some applications of marginalization relate to Model 3. In Karlsson et al. (2003), the marginalized particle filter was applied to underwater navigation using a model corresponding to (18), save the fact that $G_t^n = I$, $G_t^l = I$, $f_t^l = 0$, $A_t^n = 0$. In Gustafsson et al. (2002), a model corresponding to linear state equations and a nonlinear measurement equation is applied to various problems, such as car positioning, terrain navigation, and target tracking. Due to its relevance, this model will be discussed in more detail in Section 3. Another special case of Model 3 has been applied to problems in communication theory in Chen et al. (2000), Wang et al. (2002). The model used there is linear. However, depending on an indicator variable the model changes. Hence, this indicator variable can be thought of as the nonlinear state variable in Model 3. A good and detailed explanation of how to use the marginalized particle filter for this case can be found in Doucet et al. (2001b). They refer to the model as a jump Markov linear system.

Analogously to what has been done in (7), the filtering distribution $p(x_t|Y_t)$ is split according to

$$p(x_t^l, X_t^n|Y_t) = p(x_t^l|X_t^n, Y_t) p(X_t^n|Y_t). \quad (20)$$

The linear state variables are estimated using the Kalman filter in a slightly more general setting than which was previously discussed. However, it is still the same three steps that are executed in order to estimate the linear state variables. The first step is a measurement update using the information available in y_t . The second step is a measurement update using the information available in $\hat{x}_{t+1|t}^n$, and finally, there is a time update. The following theorem explains how the linear state variables are estimated.

Theorem A.1

Using Model 3 the conditional probability density functions for x_t^l and x_{t+1}^l are given by

$$p(x_t^l|X_t^n, Y_t) = \mathcal{N}\left(x_t^l | \hat{x}_{t|t}^l, P_{t|t}\right), \quad (21a)$$

$$p(x_{t+1}^l|X_{t+1}^n, Y_t) = \mathcal{N}\left(x_{t+1}^l | \hat{x}_{t+1|t}^l, P_{t+1|t}\right), \quad (21b)$$

where

$$\hat{x}_{t|t}^l = \hat{x}_{t|t-1}^l + K_t \left(y_t - h_t - C_t \hat{x}_{t|t-1}^l \right), \quad (22a)$$

$$P_{t|t} = P_{t|t-1} - K_t M_t K_t^T, \quad (22b)$$

$$M_t = C_t P_{t|t-1} C_t^T + R_t, \quad (22c)$$

$$K_t = P_{t|t-1} C_t^T M_t^{-1}, \quad (22d)$$

and

$$\hat{x}_{t+1|t}^l = \bar{A}_t^l \hat{x}_{t|t}^l + G_t^l (Q_t^{ln})^T (G_t^n Q_t^n)^{-1} z_t + f_t^l + L_t \left(z_t - A_t^n \hat{x}_{t|t}^l \right), \quad (23a)$$

$$P_{t+1|t} = \bar{A}_t^l P_{t|t} (\bar{A}_t^l)^T + G_t^l \bar{Q}_t^l (G_t^l)^T - L_t N_t L_t^T, \quad (23b)$$

$$N_t = A_t^n P_{t|t} (A_t^n)^T + G_t^n Q_t^n (G_t^n)^T, \quad (23c)$$

$$L_t = \bar{A}_t^l P_{t|t} (A_t^n)^T N_t^{-1}, \quad (23d)$$

where

$$z_t = x_{t+1}^n - f_t^n, \quad (24a)$$

$$\bar{A}_t^l = A_t^l - G_t^l (Q_t^{ln})^T (G_t^n Q_t^n)^{-1} A_t^n, \quad (24b)$$

$$\bar{Q}_t^l = Q_t^l - (Q_t^{ln})^T (Q_t^n)^{-1} Q_t^{ln}. \quad (24c)$$

Proof: See the Appendix. □

It is worth noting that if the cross-covariance Q_t^{ln} between the two noise sources w_t^n and w_t^l is zero, then $\bar{A}_t^l = A_t^l$ and $\bar{Q}_t^l = Q_t^l$. The first density $p(x_t^l|X_t^n, Y_t)$ on the right-hand side in (20) is now taken care of. In order for the estimation to work the second density $p(X_t^n|Y_t)$ in (20) is taken care of according to (12). The analytical expressions for $p(y_t|X_t^n, Y_{t-1})$ and $p(x_t^n|X_{t-1}^n, Y_{t-1})$ are provided by the following theorem.

Theorem A.2

For Model 3, $p(y_t|X_t^n, Y_{t-1})$ and $p(x_{t+1}^n|X_t^n, Y_t)$ are given by

$$p(y_t|X_t^n, Y_{t-1}) = \mathcal{N}\left(y_t | h_t + C_t \hat{x}_{t|t-1}^l, C_t P_{t|t-1} C_t^T + R_t\right), \quad (25a)$$

$$p(x_{t+1}^n|X_t^n, Y_t) = \mathcal{N}\left(x_{t+1}^n | f_t^n + A_t^n \hat{x}_{t|t}^l, A_t^n P_{t|t} (A_t^n)^T + G_t^n Q_t^n (G_t^n)^T\right). \quad (25b)$$

Proof: For the basic facts about conditionally linear models, see Harvey (1989). The details for this particular case can be found in Schön (2003). \square

The details for estimating the states in Model 3 have now been derived, and the complete algorithm is Algorithm A.1. As pointed out before, the only difference between this algorithm and the standard particle filtering algorithm is that the prediction stage is different. If steps 4(a) and 4(c) are removed from Algorithm A.1, the standard particle filter algorithm is obtained.

In this paper, the most basic form of the particle filter has been used. Several more refined variants exist, which in certain applications can give better performance. However, since the aim of this paper is to communicate the idea of marginalization in a general linear/nonlinear state-space model, the standard particle filter has been used. It is straightforward to adjust the algorithm given in this paper to accommodate, e.g., the auxiliary particle filter (Pitt and Shephard, 1999) and the Gaussian particle filter (Kotecha and Djuric, 2003a,b). Several ideas are also given in the papers collected in Doucet et al. (2001a).

The estimates as expected means of the linear state variables and their covariances are given by Nordlund (2002)

$$\hat{x}_{t|t}^l = \sum_{i=1}^N \tilde{q}_t^{(i)} \hat{x}_{t|t}^{l,(i)}, \quad (26a)$$

$$\hat{P}_{t|t} = \sum_{i=1}^N \tilde{q}_t^{(i)} \left(P_{t|t}^{(i)} + \left(\hat{x}_{t|t}^{l,(i)} - \hat{x}_{t|t}^l \right) \left(\hat{x}_{t|t}^{l,(i)} - \hat{x}_{t|t}^l \right)^T \right), \quad (26b)$$

where $\tilde{q}_t^{(i)}$ are the normalized importance weights, provided by step 2 in Algorithm A.1.

3 Important Special Cases and Extensions

Model 3 is quite general indeed and in most applications, special cases of it are used. This fact, together with some extensions, will be the topic of this section.

The special cases are just reductions of the general results presented in the previous section. However, they still deserve some attention in order to highlight some important mechanisms. It is worth mentioning that linear sub-structures can enter the model more implicitly as well, for example, by modeling colored noise and by sensor offsets and trends. These modeling issues are treated in several introductory texts on Kalman filtering, see, e.g., (Gustafsson, 2000, Section 8.2.4). In the subsequent section, some noise modeling aspects are discussed. This is followed by a discussion of a model with linear state equations and a nonlinear measurement equation.

3.1 Generalized Noise Assumptions

The Gaussian noise assumption can be relaxed in two special cases. First, if the measurement equation (18c) does not depend on the linear state variables x_t^l , i.e., $C_t(x_t^n) = 0$, the measurement noise can be arbitrarily distributed. In this case (18c) does not contain any information about the linear state variables, and hence, cannot be used in the Kalman filter. It is solely used in the particle filter part of the algorithm, which can handle all probability density functions.

Second, if $A_t^n(x_t^n) = 0$ in (18a), the nonlinear state equation will be independent of the linear states and, hence, cannot be used in the Kalman filter. This means that the state noise w_t^n can be arbitrarily distributed.

The noise covariances can depend on the nonlinear state variables, i.e., $R_t = R_t(x_t^n)$ and $Q_t = Q_t(x_t^n)$. This is useful for instance in terrain navigation, where the nonlinear state variable includes information about the position. Using the horizontal position and a geographic information system (GIS) on board the aircraft, noise covariances depending on the characteristics of the terrain at the current horizontal position can be motivated. We will elaborate on this issue in Section 5.

3.2 Important Model Class

A quite important special case of Model 3 is a model with linear state equations and a nonlinear measurement equation. In Model 4 below, such a model is defined.

Model 4

$$x_{t+1}^n = A_{n,t}^n x_t^n + A_{l,t}^n x_t^l + G_t^n w_t^n, \quad (27a)$$

$$x_{t+1}^l = A_{n,t}^l x_t^n + A_{l,t}^l x_t^l + G_t^l w_t^l, \quad (27b)$$

$$y_t = h_t(x_t^n) + e_t, \quad (27c)$$

with $w_t^n \sim \mathcal{N}(0, Q_t^n)$ and $w_t^l \sim \mathcal{N}(0, Q_t^l)$. The distribution for e_t can be arbitrary, but it is assumed known.

The measurement equation (27c) does not contain any information about the linear state variable x_t^l . Hence, as far as the Kalman filter is concerned, (27c) cannot be used in estimating the linear states. Instead all information from the measurements enter the Kalman filter implicitly via the second measurement update using the nonlinear state equation (27a) and the prediction of the nonlinear state $\hat{x}_{t+1|t}^n$, as a measurement. This means that in Algorithm A.1, step 4(a) can be left out. In this case, the second measurement update is much more than just a correction to the first measurement update. It is the only way in which the information in y_t enters the algorithm.

Model 4 is given special attention as several important state estimation problems can be modeled in this way. Examples include positioning, target tracking and collision avoidance (Gustafsson et al., 2002, Bar-Shalom and Li, 1993). For more information on practical matters concerning modeling issues, see, e.g., Li and Jilkov (2001, 2003), Bar-Shalom and Li (1993), Nordlund (2002). In the applications mentioned above, the nonlinear state variable x_t^n usually corresponds to the position, whereas the linear state variable x_t^l corresponds to velocity, acceleration, and bias terms.

If Model 4 is compared to Model 3, it can be seen that the matrices A_t^n, A_t^l, G_t^n , and G_t^l are independent of x_t^n in Model 4, which implies that

$$P_{t|t}^{(i)} = P_{t|t}, \quad \forall i = 1, \dots, N. \quad (28)$$

This follows from (23b) – (23d) in Theorem A.1. According to (28) only one instead of N Riccati recursions is needed, which leads to a substantial reduction in computational complexity. This is, of course, very important in real-time implementations. A further study of the computational complexity of the marginalized particle filter can be found in Karlsson et al. (2004).

If the dynamics in (18a) – (18b) are almost linear, it can be linearized to obtain a model described by (27a) – (27b). Then, the extended Kalman filter can be used instead of the Kalman filter. As is explained in Li and Jilkov (2003, 2001) it is common that the system model is almost linear, whereas the measurement model is severely nonlinear. In these cases, use the particle filter for the severe nonlinearities and the extended Kalman filter for the mild nonlinearities.

4 Illustrating Example

In order to make things as simple as possible, the following two dimensional model will be used:

$$x_{t+1} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} x_t + w_t, \quad (29a)$$

$$y_t = h(z_t) + e_t, \quad (29b)$$

where the state vector is $x_t = (z_t \ z_t)^T$. Hence, the state consists of a physical variable and its derivative. Models of this kind are very common in applications. One example is bearings-only tracking, where the objective is to estimate the angle and angular velocity and the nonlinear measurement depends on the antenna diagram. Another common application is state estimation in a DC-motor, where the angular position is assumed to be measured nonlinearly. As a final application terrain navigation in one dimension is mentioned, where the measurement is given by a map. A more realistic terrain navigation example is discussed in Section 5.

Model (29) is linear in \dot{z}_t and nonlinear in z_t . The state vector can thus be partitioned as $x_t = (x_t^n \ x_t^l)^T$, which implies that (29) can be written as

$$x_{t+1}^n = x_t^n + T x_t^l + w_t^n, \quad (30a)$$

$$x_{t+1}^l = x_t^l + w_t^l, \quad (30b)$$

$$y_t = h_t(x_t^n) + e_t, \quad (30c)$$

This corresponds to the triangular model given in Model 2. Hence, the Kalman filter for the linear state variable is given by (22) – (24), where the nonlinear state is provided by the particle filter. The estimate of the linear state variable is given by (23a), which for this example, is

$$\hat{x}_{t+1|t}^l = (1 - l_t T) \hat{x}_{t|t}^l + l_t T \frac{x_{t+1}^n - x_t^n}{T}, \quad (31)$$

where

$$n_t = T^2 p_{t|t} + q_t^n, \quad l_t = \frac{T}{n_t} p_{t|t}. \quad (32)$$

Intuitively, (31) makes sense, since the velocity estimate is given as a weighted average of the current velocity and the estimated momentary velocity, where the weights are computed from the Kalman filter quantities. In cases where (29a) is motivated by Newton's force law the unknown force is modeled as a disturbance, and $q_t^n = 0$. This implies that (31) is reduced to

$$\hat{x}_{t+1|t}^l = \frac{x_{t+1}^n - x_t^n}{T}. \quad (33)$$

Again, this can intuitively be understood, since, because it is conditioned on the knowledge of the nonlinear state variable, (30a) can be written as

$$x_t^l = \frac{x_{t+1}^n - x_t^n}{T}. \quad (34)$$

Thus, (30b) does not add any information for the Kalman filter since x_t^l is a deterministic function of the known nonlinear state variable.

5 Integrated Aircraft Navigation

As was explained in the introduction, the integrated navigation system in the Swedish fighter aircraft Gripen consists of an inertial navigation system (INS), a terrain-aided positioning (TAP) system, and an integration filter. This filter fuses the information from INS with the information from TAP; see Figure 1. The currently used integration filter is

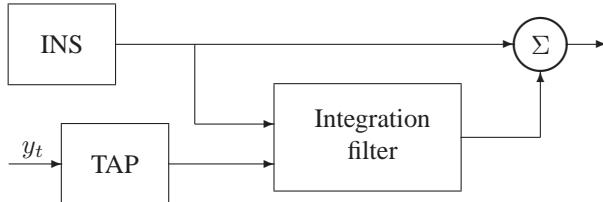


Figure 1: The integrated navigation system consists of an inertial navigation system (INS), a terrain-aided positioning (TAP) system and an integration filter. The integration filter fuse the information from INS with the information from TAP.

likely to be changed to a marginalized particle filter in the future for Gripen; see Figure 2. A first step in this direction was taken in Gustafsson et al. (2002), where a six-dimensional model was used for integrated navigation. In six dimensions, the particle filter is possible to use, but better performance can be obtained. As demonstrated in Gustafsson et al. (2002), 4000 particles in the marginalized filter outperforms 60000 particles in the standard particle filter.

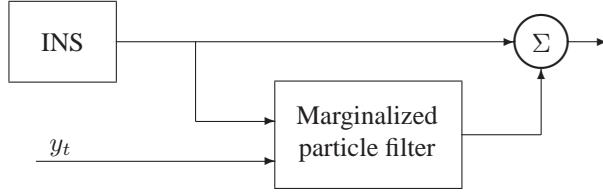


Figure 2: Using the marginalized particle filter for navigation. The terrain information is now incorporated directly in the marginalized particle filter. The radar altimeter delivers the height measurement y_t .

The feasibility study presented here applies marginalization to a more realistic nine-dimensional sub-model of the total integrated navigation system. Already here, the dimensionality has proven to be too large for the particle filter to be applied directly. The example contains all ingredients of the total system, and the principle is scalable to the full 27-dimensional state vector. The model can be simulated and evaluated in a controlled fashion; see Nordlund (2002) for more details. In the subsequent sections, the results from field trials are presented.

5.1 Dynamic Model

In order to apply the marginalized particle filter to the navigation problem a dynamic model of the aircraft is needed. In this paper the overall structure of this model is discussed. For details, see Nordlund (2002) and the references therein. The errors in the states are estimated instead of the absolute states. The reason is that the dynamics of the errors are typically much slower than the dynamics of the absolute states. The model has the following structure:

$$x_{t+1}^n = A_{n,t}^n x_t^n + A_{l,t}^n x_t^l + G_t^n w_t^n, \quad (35a)$$

$$x_{t+1}^l = A_{n,t}^l x_t^n + A_{l,t}^l x_t^l + G_t^l w_t^l, \quad (35b)$$

$$y_t = h \left(\begin{pmatrix} L_t \\ l_t \end{pmatrix} + x_t^n \right) + e_t. \quad (35c)$$

There are seven linear states, and two nonlinear states. The linear states consist of two velocity states and three states for the aircraft in terms of heading, roll, and pitch. Finally, there are two states for the accelerometer bias. The nonlinear states correspond to the error in the horizontal position, which is expressed in latitude L_t and longitude l_t .

The total dimension of the state vector is thus nine, which is too large to be handled by the particle filter. The highly nonlinear nature of measurement equation (35c), due to the terrain elevation database, implies that an extended Kalman filter cannot be used. However, the model described by (35) clearly fits into the framework of the marginalized particle filter.

The measurement noise in (35c) deserves some special attention. The radar altimeter, which is used to measure the ground clearance, interprets any echo as the ground. This is a problem when flying over trees. The tree tops will be interpreted as the ground, with

a false measurement as a result. One simple, but effective, solution to this problem is to model the measurement noise as

$$p_{e_t}(e_t) = \pi\mathcal{N}(e_t | m_1, \sigma_1) + (1 - \pi)\mathcal{N}(e_t | m_2, \sigma_2), \quad (36)$$

where π is the probability of obtaining an echo from the ground, and $(1 - \pi)$ is the probability of obtaining an echo from the tree tops. The probability density function (36) is shown in Figure 3. Experiments have shown that this, in spite of its simplicity, is a quite

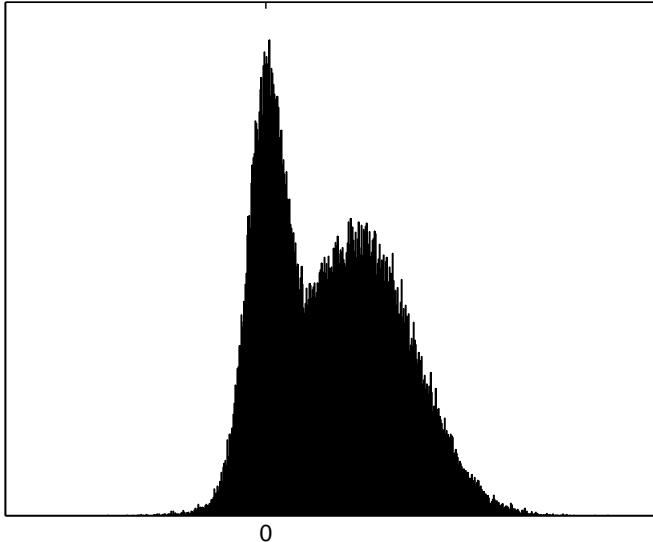


Figure 3: A typical histogram of the error in the data from the radar altimeter. The first peak corresponds to the error in the ground reading, and the second peak corresponds to the error in the readings from the tree tops.

accurate model (Dahlgren, 1998). Furthermore, m_1 , m_2 , σ_1 , σ_2 , and π in (36) can be allowed to depend on the current horizontal position L_t , l_t . In this way, information from the terrain elevation database can be inferred on the measurement noise in the model. Using this information, it is possible to model whether the aircraft is flying over open water or over a forest.

5.2 Result

The flight that has been used is shown in Figure 4. This is a fairly tough flight for the algorithm, in the sense that during some intervals data are missing, and sometimes, the radar altimeter readings become unreliable. This happens at high altitudes and during sharp turns (large roll angle), respectively. In order to get a fair understanding of the algorithm's performance, 100 Monte Carlo simulations with the same data have been performed, where only the noise realizations have been changed from one simulation to the other. Many parameters have to be chosen, but only the number of particles used

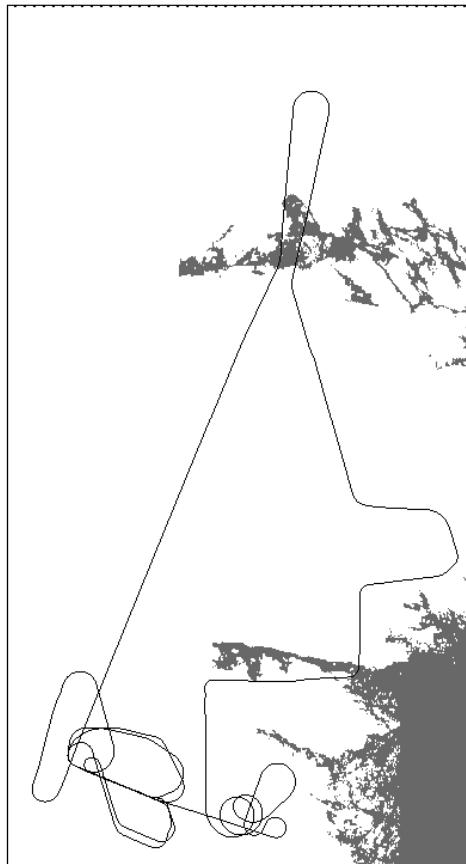


Figure 4: The flight path used for testing the algorithm. The flight path is clockwise, and the dark regions in the figure are open water.

are commented here (see Frykman (2003) for more details). In Figure 5, a plot of the error in horizontal position as a function of time is presented for different number of particles. The true position is provided by the differential GPS (DGPS). From this figure, it is obvious that the estimate improves as more particles are used. This is natural since the theory states that the densities are approximated better the more particles used. The difference in performance is mainly during the transient, where it can be motivated to use more particles. By increasing the number of particles the convergence time is significantly reduced, and a better estimate is obtained. This is true up to 5000 particles. Hence, 5000 particles were used in this study. The algorithm can be further improved, and in Frykman (2003), several suggestions are given.

The conclusion from this study is that the marginalized particle filter performs well and provides an interesting and powerful alternative to methods currently used in integrated aircraft navigation systems.

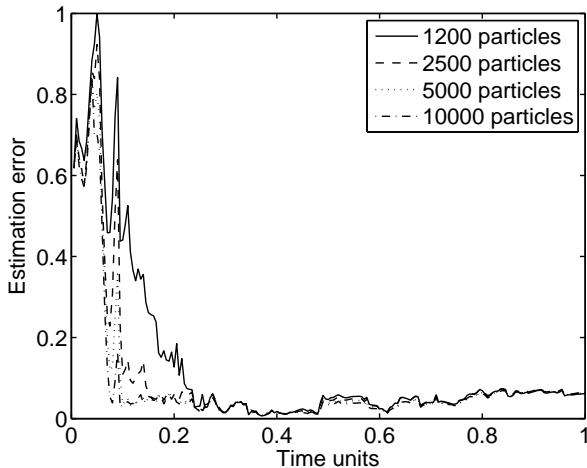


Figure 5: The horizontal position error as a function of time units for different numbers of particles. The marginalized particle filter given in Algorithm A.1 has been used.

6 Conclusion

The marginalization techniques have systematically been applied to general nonlinear and non-Gaussian state-space models, with linear sub-structures. This has been done in several steps, where each step implies a certain modification of the standard particle filter. The first step was to associate one Kalman filter with each particle. These Kalman filters were used to estimate the linear states. The second step was to use the prediction of the nonlinear state as an additional measurement. This was used to obtain better estimates of the linear state variables. The complete details for the marginalized particle filter were derived for a general nonlinear and non-Gaussian state-space model. Several important special cases were also described. Conditions implying that all the Kalman filters will obey the same Riccati recursion were given.

Finally, a terrain navigation application with real data from the Swedish fighter aircraft Gripen was presented. The particle filter is not a feasible algorithm for the full nine-state model since a huge number of particles would be needed. However, since only two states (the aircraft's horizontal position) appear nonlinearly in the measurement equation, a special case of the general marginalization algorithm can be applied. A very good result can be obtained with only 5000 particles, which is readily possible to implement in the computer currently used in the aircraft.

Acknowledgment

The authors would like to thank P. Frykman for providing Figure 5 in Section 5, which was generated as a part of his Master's thesis at Saab Aerospace, Linköping, Sweden.

The authors would also like to thank the reviewers and the editor for their detailed and constructive comments on this paper.

Appendix

A Proof for Theorem A.1

The proof of (16) and (17) is provided as a special case of the proof below.

Proof: For the sake of notational brevity, the dependence on x_t^n in (18) is suppressed in this proof. Write (18) as

$$x_{t+1}^l = f_t^l + A_t^l x_t^l + G_t^l w_t^l, \quad (37a)$$

$$z_t^1 = A_t^n x_t^l + G_t^n w_t^n, \quad (37b)$$

$$z_t^2 = C_t x_t^l + e_t, \quad (37c)$$

where z_t^1 and z_t^2 are defined as

$$z_t^1 = x_{t+1}^n - f_t^n, \quad (37d)$$

$$z_t^2 = y_t - h_t. \quad (37e)$$

Inspection of the above equations gives that z_t^1 and z_t^2 can both be thought of as measurements, since mathematically (37b) and (37c) possess the structure of measurement equations. The fact that there is a cross-correlation between the two noise processes w_t^l and w_t^n , since $Q_t^{ln} \neq 0$ in (19a), has to be taken care of. This can be accomplished using the Gram–Schmidt procedure to de-correlate the noise (Gustafsson, 2000, Kailath et al., 2000). Instead of w_t^l , the following can be used

$$\bar{w}_t^l = w_t^l - E\{w_t^l(w_t^n)^T\}(E\{w_t^n(w_t^n)^T\})^{-1}w_t^n = w_t^l - Q_t^{ln}(Q_t^n)^{-1}w_t^n, \quad (38)$$

resulting in $E\{\bar{w}_t^l(\bar{w}_t^l)^T\} = 0$ and

$$\bar{Q}_t^l = E\{\bar{w}_t^l(\bar{w}_t^l)^T\} = Q_t^l - Q_t^{ln}(Q_t^n)^{-1}Q_t^{ln}. \quad (39)$$

Using (37b) and (38), (37a) can be rewritten according to (G_t^n is assumed invertible. The case of a non-invertible G_t^n is treated in Bergman (1999))

$$\begin{aligned} x_{t+1}^l &= A_t^l x_t^l + G_t^l (\bar{w}_t^l + Q_t^{ln}(Q_t^n)^{-1}(G_t^n)^{-1}(z_t^1 - A_t^n x_t^l)) + f_t^l, \\ &= \bar{A}_t^l x_t^l + G_t^l \bar{w}_t^l + G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1} z_t^1 + f_t^l, \end{aligned} \quad (40)$$

where

$$\bar{A}_t^l = A_t^l - G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1} A_t^n. \quad (41)$$

The de-correlated system is

$$x_{t+1}^l = f_t^l + \bar{A}_t^l x_t^l + G_t^l Q_t^{ln} (G_t^n Q_t^n)^{-1} z_t^1 + G_t^l \bar{w}_t^l, \quad (42a)$$

$$z_t^1 = A_t^n x_t^l + G_t^n w_t^n, \quad (42b)$$

$$z_t^2 = C_t x_t^l + e_t, \quad (42c)$$

which is a linear system with Gaussian noise. Moreover, from (37d) and (37e), it can be seen that Z_t^1 and Z_t^2 are known if X_{t+1}^n and Y_t are known. The actual proof, using induction, of the theorem can now be started. At time zero, $p(x_0^l | X_0^n, Y_{-1}) = p(x_0^l | x_0^n) = \mathcal{N}(x_0^l | \bar{x}_0^l, \bar{P}_0)$. Now, assume that $p(x_t^l | X_t^n, Y_{t-1})$ is Gaussian at an arbitrary time, t .

The recursions are divided into three parts. First, the information available in the actual measurement y_t , i.e., z_t^2 is used. Once the measurement update has been performed the estimates $\hat{x}_{t|t}^l$ and $P_{t|t}$ are available. These can now be used to calculate the predictions of the nonlinear state $\hat{x}_{t+1|t}^n$. These predictions will provide new information about the system. Second, this new information is incorporated by performing a second measurement update using the artificial measurement z_t^1 . Finally, a time update, using the result from the second step, is performed.

Part 1: Assume that both $p(x_t^l | X_t^n, Y_{t-1}) = \mathcal{N}(x_t^l | \hat{x}_{t|t-1}^l, P_{t|t-1})$ and z_t^2 are available. This means that $p(x_t^l | X_t^n, Y_t)$ can be computed as

$$p(x_t^l | X_t^n, Y_t) = \frac{p(y_t | x_t^n, x_t^l) p(x_t^l | X_t^n, Y_{t-1})}{\int p(y_t | x_t^n, x_t^l) p(x_t^l | X_t^n, Y_{t-1}) dx_t^l}. \quad (43)$$

Using the fact that the measurement noise and, thereby, $p(y_t | x_t^n, x_t^l)$ is Gaussian and the Kalman filter it can be seen that $p(x_t^l | X_t^n, Y_t) = \mathcal{N}(x_t^l | \hat{x}_{t|t}^l, P_{t|t})$, where

$$\hat{x}_{t|t}^l = \hat{x}_{t|t-1}^l + K_t (z_t^2 - C_t \hat{x}_{t|t-1}^l), \quad (44a)$$

$$P_{t|t} = P_{t|t-1} - K_t M_t K_t^T, \quad (44b)$$

$$K_t = P_{t|t-1} C_t^T M_t^{-1}, \quad (44c)$$

$$M_t = C_t P_{t|t-1} C_t^T + R_t. \quad (44d)$$

Part 2: At this stage, z_t^1 becomes available. Use

$$p(x_t^l | X_{t+1}^n, Y_t) = \frac{p(x_{t+1}^n | x_t^n, x_t^l) p(x_t^l | X_t^n, Y_t)}{\int p(x_{t+1}^n | x_t^n, x_t^l) p(x_t^l | X_t^n, Y_t) dx_t^l} \quad (45)$$

analogously to part 1 $p(x_t^l | X_{t+1}^n, Y_t) = \mathcal{N}(x_t^l | \hat{x}_{t|t}^{l*}, P_{t|t}^*)$, where

$$\hat{x}_{t|t}^{l*} = \hat{x}_{t|t}^l + L_t (z_t^1 - A_t^n \hat{x}_{t|t}^l), \quad (46a)$$

$$P_{t|t}^* = P_{t|t} - L_t N_t^* L_t^T, \quad (46b)$$

$$L_t = P_{t|t} (A_t^n)^T (N_t^*)^{-1}, \quad (46c)$$

$$N_t^* = A_t^n P_{t|t} (A_t^n)^T + G_t^n Q_t^n (G_t^n)^T. \quad (46d)$$

Part 3: The final part is the time update, i.e., to compute

$$p(x_{t+1}^l | X_{t+1}^n, Y_t) = \int p(x_{t+1}^l | x_{t+1}^n, x_t^n, x_t^l) p(x_t^l | X_{t+1}^n, Y_t) dx_t^l. \quad (47)$$

Since the state noise is Gaussian, this corresponds to the time update handled by the Kalman filter. Hence, $p(x_{t+1}^l | X_{t+1}^n, Y_t) = \mathcal{N}(x_{t+1}^l | \hat{x}_{t+1|t}^l, P_{t+1|t})$, where

$$\hat{x}_{t+1|t}^l = \bar{A}_t^l \hat{x}_{t|t}^l + G_t^l (Q_t^{ln})^T (G_t^n Q_t^n)^{-1} z_t^1 + f_t^l + L_t (z_t^1 - A_t^n \hat{x}_{t|t}^l), \quad (48a)$$

$$P_{t+1|t} = \bar{A}_t^l P_{t|t} (\bar{A}_t^l)^T + G_t^l \bar{Q}_t^l (G_t^l)^T - L_t N_t L_t^T, \quad (48b)$$

$$L_t = \bar{A}_t^l P_{t|t} (A_t^n)^T N_t^{-1}, \quad (48c)$$

$$N_t = A_t^n P_{t|t} (A_t^n)^T + G_t^n Q_t^n (G_t^n)^T. \quad (48d)$$

□

References

- Anderson, B. D. O. and Moore, J. B. (1979). *Optimal Filtering*. Information and system science series. Prentice Hall, Englewood Cliffs, NJ, USA.
- Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society*, 64(4):827–836.
- Andrieu, C. and Godsill, S. J. (2000). A particle filter for model based audio source separation. In *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000)*, Helsinki, Finland.
- Bar-Shalom, Y. and Li, X.-R. (1993). *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, Norwood, MA, USA.
- Bergman, N. (1999). *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Dissertations No 579, Linköping Studies in Science and Technology, SE-581 83 Linköping, Sweden.
- Bergman, N., Ljung, L., and Gustafsson, F. (1999). Terrain navigation using Bayesian statistics. *IEEE Control Systems Magazine*, 19(3):33–40.
- Casella, G. and Robert, C. P. (1996). Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94.
- Chen, R. and Liu, J. S. (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society*, 62(3):493–508.
- Chen, R., Wang, X., and Liu, J. S. (2000). Adaptive joint detection in flat-fading channels via mixture Kalman filtering. *IEEE Transactions on Information Theory*, 46(6):2079–2094.

- Dahlgren, C. (1998). Non-linear black box modelling of JAS 39 Gripen's radar altimeter. Master's Thesis No LiTH-ISY-EX-1958, Department of Electrical Engineering, Linköping University, Sweden.
- Doucet, A., de Freitas, N., and Gordon, N., editors (2001a). *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, USA.
- Doucet, A., Godsill, S. J., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.
- Doucet, A., Gordon, N., and Krishnamurthy, V. (1999). Particle filters for state estimation of jump Markov linear systems. Technical Report CUED/F-INFENG/TR 359, Signal Processing Group, Department of Engineering, University of Cambridge, Trupington street, CB2 1PZ Cambridge.
- Doucet, A., Gordon, N., and Krishnamurthy, V. (2001b). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624.
- Frykman, P. (2003). Applied particle filters in integrated aircraft navigation. Master's Thesis No LiTH-ISY-EX-3406, Department of Electrical Engineering, Linköping University, Sweden.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113.
- Gustafsson, F. (2000). *Adaptive Filtering and Change Detection*. John Wiley & Sons, New York, USA.
- Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J. (2002). Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437.
- Harvey, A. C. (1989). *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press, Cambridge, UK.
- Jazwinski, A. H. (1970). *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York, USA.
- Kailath, T., Sayed, A. H., and Hassibi, B. (2000). *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, NJ, USA.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45.
- Karlsson, R. and Gustafsson, F. (2003). Particle filter for underwater navigation. In *Proceedings of the Statistical Signal Processing Workshop*, pages 509–512, St. Louis, MO, USA.

- Karlsson, R., Gustafsson, F., and Karlsson, T. (2003). Particle filter and Cramér-Rao lower bound for underwater navigation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong.
- Karlsson, R., Schön, T., and Gustafsson, F. (2004). Complexity analysis of the marginalized particle filter. Technical Report LiTH-ISY-R-2611, Department of Electrical Engineering, Linköping University.
- Kotecha, J. H. and Djuric, P. M. (2003a). Gaussian particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2592–2601.
- Kotecha, J. H. and Djuric, P. M. (2003b). Gaussian sum particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2602–2610.
- Li, P., Goodall, R., and Kadirkamanathan, V. (2003). Parameter estimation of railway vehicle dynamic model using Rao-Blackwellised particle filter. In *Proceedings of the European Control Conference*, Cambridge, UK.
- Li, X. R. and Jilkov, V. P. (2001). A survey of maneuvering target tracking-part III: Measurement models. In *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, pages 423–446, San Diego, USA.
- Li, X. R. and Jilkov, V. P. (2003). Survey of maneuvering target tracking-part I: dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364.
- Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer, New York, USA.
- Nordlund, P.-J. (2002). *Sequential Monte Carlo Filters and Integrated Navigation*. Licentiate Thesis No 945, Department of Electrical Engineering, Linköping University, Sweden.
- Palmqvist, J. (1997). *On Integrity Monitoring of Integrated Navigation Systems*. Licentiate Thesis No 600, Department of Electrical Engineering, Linköping University, Sweden.
- Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599.
- Schön, T. (2003). *On Computational Methods for Nonlinear Estimation*. Licentiate Thesis No 1047, Department of Electrical Engineering, Linköping University, Sweden.
- Schön, T. and Gustafsson, F. (2003). Particle filters for system identification of state-space models linear in either parameters or states. In *Proceedings of the 13th IFAC Symposium on System Identification*, pages 1287–1292, Rotterdam, The Netherlands. Invited paper.
- Wang, X., R. Chen, R., and Guo, D. (2002). Delayed-pilot sampling for mixture Kalman filter with application in fading channels. *IEEE Transactions on Signal Processing*, 50(2):241–254.

Paper B

Complexity Analysis of the Marginalized Particle Filter

Authors: Rickard Karlsson, Thomas Schön, and Fredrik Gustafsson.

Edited version of paper originally published in *IEEE Transactions on Signal Processing*, 53(11), pages 4408–4411, November 2005.

Preliminary version published as Technical Report LiTH-ISY-R-2611, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Complexity Analysis of the Marginalized Particle Filter

Rickard Karlsson, Thomas Schön, and Fredrik Gustafsson

Department of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden.
E-mail: (rickard, schon, fredrik)@isy.liu.se.

Abstract

In this paper, the computational complexity of the marginalized particle filter is analyzed and a general method to perform this analysis is given. The key is the introduction of the *equivalent flop* measure. In an extensive Monte Carlo simulation, different computational aspects are studied and compared with the derived theoretical results.

Keywords: Complexity analysis, equivalent flop, Kalman filter, marginalized particle filter, nonlinear estimation.

1 Introduction

IN particle filter (PF) applications, knowledge of the computational complexity is often of paramount importance. In this paper the computational complexity issues that arise in the use of the *marginalized particle filter* (MPF), also called the Rao-Blackwellized particle filter are studied. The MPF is a clever combination of the standard PF (Gordon et al., 1993), and the *Kalman filter* (KF) (Kalman, 1960), which can be used when the model contains a linear sub-structure, subject to Gaussian noise. It is a well-known fact that in some cases it is possible to obtain better estimates, i.e., estimates with reduced variance, using the MPF instead of using the standard PF (Doucet et al., 2001b). By now, quite a lot has been written about the MPF, see, e.g., Doucet et al. (2000, 2001a), Chen and Liu (2000), Andrieu and Doucet (2002), Andrieu and Godsill (2000), Schön et al. (2005). However, to the best of the author's knowledge, nothing has yet been written about complexity issues. In this paper, expressions for the complexity $\mathcal{C}(p, k, N)$ are derived, where p and k represent the state dimensions from the PF and the KF, respectively and, N denotes the number of particles. A general method to analyze the computational complexity of the MPF will be provided. The method is illustrated using a common tracking model, but can be applied to a much broader class of models. For more details of the proposed method, see Karlsson et al. (2004).

2 Marginalized Particle Filter

Many nonlinear estimation problems can be handled using the particle filter. A general state-space model

$$x_{t+1} = f(x_t, w_t), \quad (1a)$$

$$y_t = h(x_t, e_t), \quad (1b)$$

has both nonlinear dynamics f and nonlinear measurements h . The noise processes w_t and e_t have known probability density functions. If the state-space model contains a linear-Gaussian sub-structure, this can be exploited to obtain better estimates using the MPF. In this paper, the case with linear-Gaussian dynamics,

$$x_{t+1} = A_t x_t + w_t, \quad w_t \in \mathcal{N}(0, Q_t), \quad (2a)$$

$$y_t = h(x_t^n) + C_t x_t^l + e_t, \quad (2b)$$

is discussed. In this context, the state variable $x_t \in \mathbf{R}^m$ is

$$x_t = \begin{pmatrix} x_t^n \\ x_t^l \end{pmatrix}, \quad (3)$$

where $x_t^l \in \mathbf{R}^l$ denotes the linear states and $x_t^n \in \mathbf{R}^n$ denotes the nonlinear states. Furthermore, $X_t^n = \{x_i^n\}_{i=0}^t$ and $Y_t = \{y_i\}_{i=0}^t$. Using Bayes' theorem,

$$p(x_t^l, X_t^n | Y_t) = p(x_t^l | X_t^n, Y_t) p(X_t^n | Y_t), \quad (4)$$

where $p(X_t^n | Y_t)$ is given by the PF and $x_t^l | X_t^n$ is linear-Gaussian, i.e., $p(x_t^l | X_t^n, Y_t)$ is given by the KF. This marginalization idea is certainly not new (Doucet et al., 2000, Casella and Robert, 1996, Doucet et al., 2001b, Chen and Liu, 2000, Andrieu and Doucet, 2002, Doucet et al., 2001b, Schön et al., 2005, Nordlund, 2002). The state vector x_t can be partitioned into two parts, $x_t^p \in \mathbf{R}^p$ and $x_t^k \in \mathbf{R}^k$, which are estimated using the PF and the KF respectively. Furthermore, $p \in [n, n+l]$, $k \in [0, l]$ and for the general partitioning case $p - n$ states can be selected from l possibilities.

It is interesting to consider which states to put in the nonlinear and the linear partition, respectively. Two relevant aspects with respect to this partitioning are how it will affect the computational complexity and the estimation performance. This will be discussed using the following model:

$$x_{t+1}^p = A_t^p x_t^p + A_t^k x_t^k + w_t^p, \quad w_t^p \sim \mathcal{N}(0, Q_t^p), \quad (5a)$$

$$x_{t+1}^k = F_t^p x_t^p + F_t^k x_t^k + w_t^k, \quad w_t^k \sim \mathcal{N}(0, Q_t^k), \quad (5b)$$

$$y_t = h_t(x_t^p) + C_t x_t^k + e_t, \quad e_t \sim \mathcal{N}(0, R_t), \quad (5c)$$

where the noise is assumed to be independent. This is no restriction, since the case of dependent noise can be reduced to the case of independent noise using a Gram–Schmidt procedure (Kailath et al., 2000). In Algorithm B.1, the MPF is summarized for the model given in (5) (with $C_t = 0$, for the sake of brevity). For a detailed derivation (including the case $C_t \neq 0$), the reader is referred to Schön et al. (2005).

Algorithm B.1 (Marginalized Particle Filter (MPF), $C_t = 0$)

1. **Initialization:** For $i = 1, \dots, N$, initialize the particles, $x_{0|-1}^{p,(i)} \sim p_{x_0^p}(x_0^p)$ and set $\{x_{0|-1}^{k,(i)}, P_{0|-1}^{(i)}\} = \{\bar{x}_0^k, \bar{P}_0\}$. Set $t := 0$.
2. **Particle filter measurement update:** For $i = 1, \dots, N$, evaluate the importance weights $q_t^{(i)} = p(y_t | X_t^{p,(i)}, Y_{t-1}) = \mathcal{N}(y_t | h(x_t^{p,(i)}), R_t)$ and normalize $\tilde{q}_t^{(i)} = q_t^{(i)} / \sum_{j=1}^N q_t^{(j)}$.
3. **Resampling:** Resample N particles with replacement according to,

$$\Pr(x_{t|t}^{p,(i)} = x_{t|t-1}^{p,(j)}) = \tilde{q}_t^{(j)}. \quad (6)$$

4. **Particle filter time update and Kalman filter update**

- (a) **Kalman filter measurement update,**

$$\hat{x}_{t|t}^{k,(i)} = \hat{x}_{t|t-1}^{k,(i)}, \quad P_{t|t} = P_{t|t-1}. \quad (7)$$

- (b) **Particle filter time update (prediction):** For $i = 1, \dots, N$,

$$x_{t+1|t}^{p,(i)} \sim p(x_{t+1|t}^p | X_t^{p,(i)}, Y_t), \quad (8)$$

where

$$p(x_{t+1}^{p,(i)} | X_t^{p,(i)}, Y_t) = \mathcal{N}\left(x_{t+1}^{p,(i)} | A_t x_t^{p,(i)} + A_t^k \hat{x}_{t|t}^{k,(i)}, A_t^k P_{t|t} (A_t^k)^T + Q_t^p\right). \quad (9)$$

- (c) **Kalman filter time update,**

$$\hat{x}_{t+1|t}^{k,(i)} = F_t^k \hat{x}_{t|t}^{k,(i)} + F_t^p x_t^{p,(i)} + L_t \left(x_{t+1|t}^{p,(i)} - A_t^p x_t^{p,(i)} - A_t^k \hat{x}_{t|t}^{k,(i)} \right), \quad (10a)$$

$$P_{t+1|t} = F_t^k P_{t|t} (F_t^k)^T + Q_t^k - L_t M_t L_t^T, \quad (10b)$$

$$M_t = A_t^k P_{t|t} (A_t^k)^T + Q_t^p, \quad (10c)$$

$$L_t = F_t^k P_{t|t} (A_t^k)^T M_t^{-1}, \quad (10d)$$

5. Set $t := t + 1$ and iterate from step 2.
-

3 Complexity Analysis

In this section the computational complexity of the MPF is discussed from a theoretical point of view, by giving the number of *floating-point operations* (flops) used in the algorithm. A flop is here defined as one addition, subtraction, multiplication, or division of two floating-point numbers. However, problems occur when the flop count is compared to the actual computation time. This is due to the fact that issues such as cache boundaries

and locality of reference will significantly influence the computation time (Boyd and Vandenberghe, 2004). Moreover, there are certain steps in the algorithm that cannot easily be measured in flops, for instance the cost of generating a random number and the cost of evaluating a nonlinear function. Despite these drawbacks, it is still possible to analyze the complexity using the computer to measure the absolute time that the different steps require. These can then be compared to the theoretical result obtained from counting flops. In the PF, the computational complexity of the resampling step is proportional to the number of particles and the amount of time for generating random numbers is proportional to the number of random numbers required. The proportionality coefficients are related to reflect the flop complexity instead of the time complexity for ease of comparison with parts that only depend on matrix and vector operations. This will be referred to as the equivalent flop (EF) complexity.

Definition B.1. *The equivalent flop (EF) complexity for an operation is defined as the number of flops that results in the same computational time as the operation.*

3.1 Nonlinear Measurements

In this section, the case $C_t = 0$ in (5c) is discussed. The total complexity of Algorithm B.1 is given for each code line in Table 1. For instance, the first instruction $P_{t|t}(A_t^k)^T$ corresponds to multiplying $P_{t|t} \in \mathbf{R}^{k \times k}$ with $(A_t^k)^T \in \mathbf{R}^{k \times p}$, which requires pk^2 multiplications and $(k - 1)kp$ additions (Golub and Van Loan, 1996). The total EF complexity is given by

$$\begin{aligned}\mathcal{C}(p, k, N) \approx & 4pk^2 + 8kp^2 + \frac{4}{3}p^3 + 5k^3 - 5kp + 2p^2 \\ & + (6kp + 4p^2 + 2k^2 + p - k + pc_3 + c_1 + c_2)N.\end{aligned}\quad (11)$$

As shown above, the coefficient c_1 has been used for the calculation of the Gaussian likelihood, c_2 for the resampling and c_3 for the random number complexity. Note that, when $C_t = 0$ the same covariance matrix is used for all Kalman filters, which reduces the computational complexity.

The analysis provided above is general and the main steps, which will be discussed in the subsequent section are as follows:

1. Estimate the time for one flop using linear regression.
2. Estimate the time for likelihood calculation, resampling and random number generation.
3. Relate all times using the EF measure.
4. Calculate the overall complexity $\mathcal{C}(p, k, N)$.

By requiring $\mathcal{C}(p + k, 0, N_{\text{PF}}) = \mathcal{C}(p, k, N(k))$, where N_{PF} corresponds to the number of particles used in the standard PF $N(k)$ can be solved for. This gives the number of particles $N(k)$ that can be used in the MPF in order to obtain the same computational complexity as if the standard particle filter had been used for all states. In Figure 1 the ratio $N(k)/N_{\text{PF}}$ is plotted for systems with $m = 3, \dots, 9$ states. Hence, using Figure 1

Table 1: The EF complexity for the PF (upper) and KF time update (lower) in Algorithm B.1 (\dagger represents the case $k > 0$, \ddagger represent operations not from matrix multiplications and additions, such as resampling, random number generation, etc.).

Instruction	Mult.	Add.	Other \ddagger
$P_A := P_{t t}(A_t^k)^T$	pk^2	$(k-1)kp$	
$M := A_t^k P_A + Q_t^p$	kp^2	$(k-1)p^2 + p^2 \dagger$	
$T_1 := chol(M)$			$\frac{p^3}{3} + 2p^2$
$T_2 := randn(p, N)$			pNc_3
$w := T_1 * T_2$	p^2N	$(p-1)pN$	
$T_3 := A^p x^p$	p^2N	$(p-1)pN$	
$T_4 := A^k x^k$	pkN	$(k-1)pN \dagger$	
$\hat{x}_{t+1 t}^p := T_3 + T_4 + w$		$2pN$	
$inv_M := M^{-1}$			p^3
$L := F_t^k P_A inv_M$	$k^2p + kp^2$	$k^2p + p^2k - 2kp$	
$T_5 := F_t^k P_{t t}(F_t^k)^T$	$2k^3$	$2(k-1)k^2$	
$T_6 := L_t M_t L_t^T$	$2kp^2$	$2(p-1)pk$	
$P := T_5 + Q_t^k - T_6$		$2k^2$	
$T_7 := F^k x^k$	k^2N	$(k-1)kN$	
$T_8 := F^p x^p$	kpN	$(p-1)kN$	
$T_9 := \hat{x}_{t+1 t}^p - T_3 - T_4$		$2pN$	
$\hat{x}_{t+1 t}^k := T_7 + T_8 + LT_9$	kpN	$(p+1)kN$	

it is possible to directly find out how much there is to gain in using the MPF from a computational complexity point of view. The figure also shows that the computational complexity is always reduced when the MPF can be used instead of the standard PF. Furthermore, it is well-known that the quality of the estimates will improve or remain the same when the MPF is used (Doucet et al., 2001b).

3.2 Mixed Nonlinear/Linear Measurements

It is now assumed that $C_t \neq 0$ in (5c), which implies that the Riccati recursions have to be evaluated for each particle. This results in a significant increase in the computational complexity. Hence, different covariance matrices have to be used for each Kalman filter, implying that (11) has to be modified. For details, see Karlsson et al. (2004), but approximately the complexity is given by

$$\begin{aligned} \mathcal{C}(p, k, N) \approx & (6kp + 4p^2 + 2k^2 + p - k + pc_3 + c_1 + c_2 \\ & + 4pk^2 + 8kp^2 + \frac{4}{3}p^3 + 5k^3 - 5kp + 2p^2 + k^3)N. \end{aligned} \quad (12)$$

The problem with the increased complexity in (12) might be reduced simply by moving one or more linear states from x_t^k to x_t^p . In Figure 2 the ratio $N(k)/N_{PF}$ is plotted for systems with $m = 3, \dots, 9$ states. For systems with few states, the MPF is more efficient than the standard PF. However, for systems with more states, where most of the states are marginalized the standard PF becomes more efficient than the MPF. The reason is the increased complexity in the Kalman filters due to the increased dimension in the Riccati

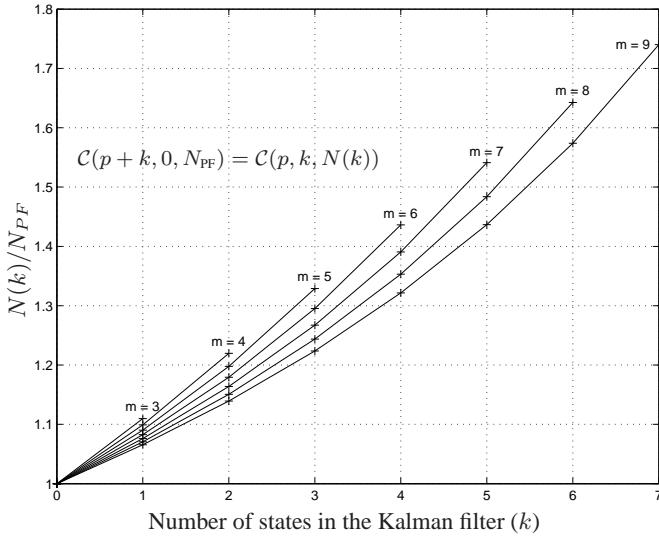


Figure 1: Ratio $N(k)/N_{PF}$ for systems with $m = 3, \dots, 9$ states and $C_t = 0, n = 2$ is shown. It is apparent the MPF can use more particles for a given computational complexity, when compared to the standard PF.

recursions. For example, according to Figure 2, a system with nine states, where seven are marginalized, $N(k) < N_{PF}$.

4 Target Tracking Example

The general method for analyzing the computational complexity presented in the previous section is illustrated using a common tracking model. The problem of estimating the position and velocity of an aircraft is studied using

$$x_{t+1} = \begin{pmatrix} 1 & 0 & T & 0 & T^2/2 & 0 \\ 0 & 1 & 0 & T & 0 & T^2/2 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} x_t + w_t, \quad (13a)$$

$$y_t = \begin{pmatrix} \sqrt{p_x^2 + p_y^2} \\ \arctan\left(\frac{p_y}{p_x}\right) \end{pmatrix} + e_t, \quad (13b)$$

where $Q = \text{Cov}(w) = \text{diag}(1 \ 1 \ 1 \ 0.01 \ 0.01)$, $R = \text{Cov}(e) = \text{diag}(100 \ 10^{-6})$, and the state vector is $x_t = (p_x \ p_y \ v_x \ v_y \ a_x \ a_y)^T$, i.e., position, velocity and acceleration. The measurement equation gives the range and azimuth from the radar system.

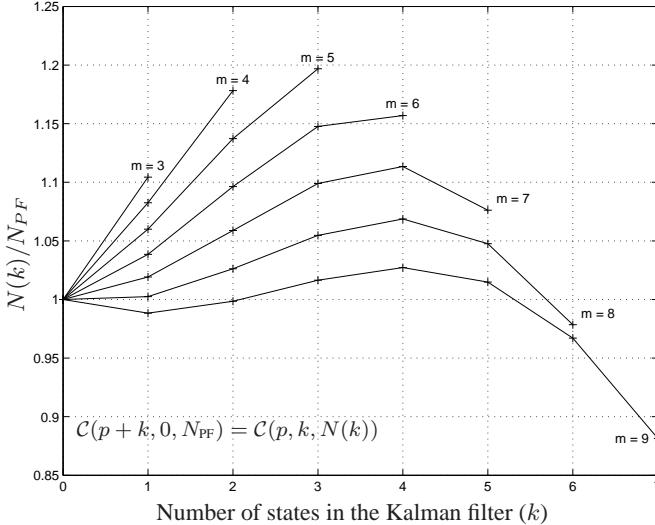


Figure 2: Ratio $N(k)/N_{PF}$ for systems with $m = 3, \dots, 9$ states and $C_t \neq 0$, $n = 2$ is shown. For systems with high state dimension and many marginalized states the standard PF can use more particles than the MPF.

In the subsequent section, a numerical study of the computational complexity is given, where the theoretical expressions previously derived are validated. Furthermore, the MPF will be analyzed in an extensive *Monte Carlo* (MC) simulation using the model described in (13). The main purpose of this simulation is to illustrate the implications of the results derived in this paper. In the simulations, one state trajectory with different noise realizations have been used. The purpose of the simulations presented here is to show that using marginalization the computational complexity is significantly reduced and the quality of the estimates is improved.

4.1 Numerical Complexity Analysis

The model (13) has two nonlinear state variables and four linear state variables, implying $k \in [0, 4]$, $p \in [2, 6]$. Two cases are now studied, the full PF, where all states are estimated using the PF and the completely marginalized PF, where all linear states are marginalized out and estimated using the KF. Requiring the same computational complexity, i.e., $\mathcal{C}(6, 0, N_{PF}) = \mathcal{C}(2, 4, N_{MPF})$, gives

$$N_{PF} = \underbrace{\left(1 - \frac{4c_3 + 56}{c_1 + c_2 + 6c_3 + 150}\right)}_{<1} N_{MPF}. \quad (14)$$

From (14), it is clear that for a given computational complexity more particles can be used in the MPF than in the standard PF. Expression (14) is a specific instance of what has been plotted in Figure 1, where the curve corresponds to $m = 6$, $k = 4$. In order to quantify this statement, numerical values for the three constants c_1 , c_2 and c_3 are needed. They are estimated by analyzing the actual computational time consumed by various parts of the MPF algorithm. It was fairly easy to measure the time used for likelihood calculation, resampling, and random number generation as a function of the number of particles. The problem is to relate them to the time consumed for a single flop. For simpler hardware implementations, one flop would have a constant execution time. However, in order to do this on a normal desktop computer running MATLAB, the EF estimation has to be considered, since flop count does not entirely reflect the actual computational time. This is due to memory caching, pipelining, efficient computational routines which are problem size dependent, and memory swapping. For the tracking example from (13) the estimated coefficients are $c_1 = 445$, $c_2 = 487$, and $c_3 = 125$ (on a Sun Blade 100 with 640 MB memory).

By comparing the EF complexity given by (11) to the actual computational time measured in MATLAB, it is clear that the predictions of the computational complexity based on theoretical considerations are quite good indeed. The result is given in Figure 3. The

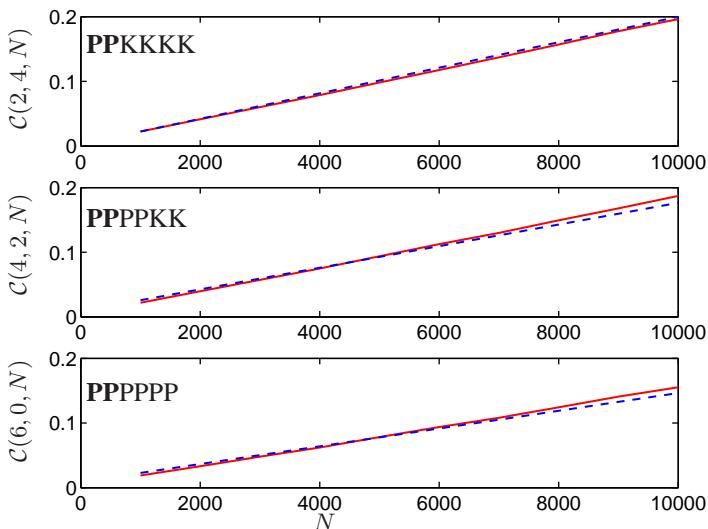


Figure 3: Using a constant number of particles the times predicted from the theoretical results are shown by the dashed line. The solid line corresponds to the actual time measured using MATLAB. If a certain state variable is estimated using the PF this is indicated with a P , and if the KF is used this is indicated using a K .

small error is mainly due to the fact that it is quite hard to predict the time used for matrix operations, as previously discussed.

4.2 Simulation – Constant Time

Using a constant time the number of particles that can be used is computed. The study is performed by first running the full PF and measure the time consumed by the algorithm. An MC simulation, using $N = 2000$ particles, is performed in order to obtain a stable estimate of the time consumed by the algorithm. To avoid intervention from the operating system, the minimum value is chosen. The time is then used as the target function for the different partitions in the MPF. To find the number of particles needed,

Table 2: Results from the constant time simulation.

	PPPPPP	PPKKPP	PPPPKK	PPKKKK
N	2000	2029	1974	2574
RMSE pos	7.10	5.81	5.76	5.60
RMSE vel	3.62	3.27	3.28	3.21
RMSE acc	0.52	0.47	0.45	0.44
Time	0.59	0.58	0.57	0.60

a search method is implemented and MC simulations are used to get a stable estimate. In Table 2, the number of particles (N), the *root mean square error* (RMSE) and simulation times are shown for the different marginalization cases. RMSE is defined as $\left(\frac{1}{T_f} \sum_{i=1}^{T_f} \frac{1}{N_{MC}} \sum_{j=1}^{N_{MC}} \|x_i^{\text{TRUE}} - \hat{x}_i^{(j)}\|_2^2 \right)^{1/2}$, where T_f is the number of time samples and $N_{MC} = 100$ is the number of MC simulations used. From Table 2, it is clear that the different MPFs can use more particles for a given time, which is in perfect correspondence with the theoretical result given in (14). From the study, it is also concluded that the RMSE is decreasing when marginalization is used. This is also in accordance with theory, which states that the variance should decrease or remain unchanged when marginalization is used (Doucet et al., 2001b). Furthermore, Table 2 verifies the theoretical results presented in Figure 1. From this figure it is also clear that the complete marginalization ($m = 6, k = 4$) gives $N(k)/N_0 = 1.44$. Hence, the theoretically predicted number of particles is $2000 \times 1.44 = 2880$. This is in quite good agreement with the result reported in table 2, 2574.

4.3 Simulation – Constant Velocity RMSE

In this section, we study what happens if a constant velocity RMSE is used. First, the velocity RMSE for the full PF is found using an MC simulation. This value is then used as a target function in the search for the number of particles needed by the different MPFs. Table 3 clearly indicates that the MPF can obtain the same RMSE using fewer particles. The result is that using full marginalization only requires 14% of the computational resources as compared to the standard PF in this example.

Table 3: Results using a constant velocity RMSE.

	PPPPPP	PPKKPP	PPPPKK	PPKKKK
<i>N</i>	2393	864	943	264
RMSE pos	7.07	6.98	7.12	7.27
RMSE vel	3.58	3.60	3.65	3.61
RMSE acc	0.50	0.51	0.49	0.48
Time	0.73	0.26	0.28	0.10

5 Conclusion

The contribution in this paper is a systematic approach to analyze the marginalized particle filter from a computational complexity point of view. The method is general and can be applied to a large class of problems. To illustrate the idea, a common target tracking problem is analyzed in detail. The complexity analysis is performed theoretically by counting the number of flops and using the equivalent flop measure to account for complex algorithmic parts such as random number generation and resampling. In an extensive Monte Carlo simulation, different performance aspects are shown, and the theoretical results are illustrated and validated.

Acknowledgment

This work was supported by VINNOVA's Center of Excellence ISIS (Information Systems for Industrial Control and Supervision), in particular the partner NIRA Dynamics, and by the Swedish Research Council (VR). The authors would also like to thank the reviewers for their constructive comments.

References

- Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society, 64(4)*:827–836.
- Andrieu, C. and Godsill, S. J. (2000). A particle filter for model based audio source separation. In *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000)*, Helsinki, Finland.
- Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge, UK.
- Casella, G. and Robert, C. P. (1996). Rao-Blackwellisation of sampling schemes. *Biometrika, 83(1)*:81–94.
- Chen, R. and Liu, J. S. (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society, 62(3)*:493–508.

- Doucet, A., de Freitas, N., and Gordon, N., editors (2001a). *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, USA.
- Doucet, A., Godsill, S. J., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.
- Doucet, A., Gordon, N., and Krishnamurthy, V. (2001b). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624.
- Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*. John Hopkins University Press, Baltimore, third edition.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113.
- Kailath, T., Sayed, A. H., and Hassibi, B. (2000). *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, NJ, USA.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45.
- Karlsson, R., Schön, T., and Gustafsson, F. (2004). Complexity analysis of the marginalized particle filter. Technical Report LiTH-ISY-R-2611, Department of Electrical Engineering, Linköping University.
- Nordlund, P.-J. (2002). *Sequential Monte Carlo Filters and Integrated Navigation*. Licentiate Thesis No 945, Department of Electrical Engineering, Linköping University, Sweden.
- Schön, T., Gustafsson, F., and Nordlund, P.-J. (2005). Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289.

Paper C

A Modeling and Filtering Framework for Linear Differential-Algebraic Equations

Authors: Thomas Schön, Markus Gerdin, Torkel Glad, and Fredrik Gustafsson.

Edited version of paper originally published in *Proceedings of the 42th IEEE Conference on Decision and Control*, pages 892 – 897, Hawaii, USA, December 2003.

An extended version of this paper has been submitted to *Automatica*.

Preliminary version published as Technical Report LiTH-ISY-R-2498, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

A Modeling and Filtering Framework for Linear Differential-Algebraic Equations

Thomas Schön, Markus Gerdin, Torkel Glad, and Fredrik Gustafsson

Department of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden.
E-mail: (schon, gerdin, torkel, fredrik)@isy.liu.se.

Abstract

General approaches to modeling, for instance using object-oriented software, lead to differential-algebraic equations (DAE). As the name reveals, it is a combination of differential and algebraic equations. For state estimation using observed system inputs and outputs in a stochastic framework similar to Kalman filtering, we need to augment the DAE with stochastic disturbances (“process noise”), whose covariance matrix becomes the tuning parameter. We will determine the subspace of possible disturbances based on the linear DAE model. This subspace determines all degrees of freedom in the filter design, and a Kalman filter algorithm is given. We illustrate the design on a system with two interconnected rotating masses.

Keywords: Differential-algebraic equations, implicit systems, singular systems, descriptor systems, white noise, noise, discretization, Kalman filter.

1 Introduction

IN recent years so-called object-oriented modeling software has increased in popularity. Examples of such software are Omola, Dymola, the SimMechanics toolbox for MATLAB, and Modelica (Mattsson et al., 1998, Tiller, 2001). Such modeling software makes it possible to model physical systems by connecting sub-models in a way which parallels the physical construction and without having to manually manipulate any equations. The available software usually gives the user the possibility to simulate the system, and perhaps also to extract a structured model in an automatic way. This model generally becomes a differential-algebraic equation (DAE), which in the linear case can be written

$$E\dot{x}(t) + Fx(t) = B_u u(t), \quad (1a)$$

where $x(t)$ is the internal variable vector, $u(t)$ is the system input vector and E, F, B_u are matrices of appropriate dimensions. We assume that E is singular, otherwise we get an ordinary differential equation (ODE) by simply multiplying with E^{-1} from the left, and the standard Kalman filtering theory applies. Hence, when E is singular we obtain a differential-algebraic equation and the reason for the singularity is often that purely algebraic equations are present. Other common names for the model structure (1a) are, e.g., implicit systems, descriptor systems, semi-state systems, generalized systems, and differential equations on a manifold (Campbell, 1990).

We have the possibility to place sensors in the system to get a measurement equation

$$y(t) = Cx(t) + e(t), \quad (1b)$$

where $y(t)$ is the measurement and $e(t)$ the sensor noise. An important special case we will discuss separately is for computer controlled systems, where the measurements $y[k]$ are available at the sampling times $t = kT_s$,

$$E\dot{x}(t) + Fx(t) = B_u u(t), \quad (2a)$$

$$y[kT_s] = Cx(kT_s) + e[kT_s]. \quad (2b)$$

The estimation problem is to estimate $x(t)$ from $y[kT_s]$. There are two reasons why we have to introduce process noise to (2a):

- There are unmodeled dynamics and disturbances acting on the system, that can only be included in the model as an unknown stochastic term.
- There is a practical need for tuning the filter in order to make a trade-off between tracking ability and sensor noise attenuation. This is in the Kalman filter accomplished by keeping the sensor noise covariance matrix constant and tuning the process noise covariance matrix, or the other way around. Often, it is easier to describe the sensor noise in a stochastic setting, and then it is more natural to tune the process noise.

With process noise, the model (1) becomes

$$E\dot{x}(t) + Fx(t) = B_u u(t) + B_w w(t), \quad (3a)$$

$$y(t) = Cx(t) + e(t). \quad (3b)$$

The problem is to determine where in the system disturbances can occur. To fit the optimal filtering and Kalman filtering framework, $w(t)$ should be white noise. As will be demonstrated, adding white noise to all equations can lead to derivatives of white noise affecting internal variables of the system directly. This will be referred to as a noncausal system, with a physical interpretation of infinite forces, currents etc. Therefore, we will derive a basis for the subspace of all possible disturbances, that leads to causal systems. This basis is taken as B_w in (3), and the process noise covariance matrix $Q = \text{Cov}\{w(t)\}$ is used as the design variable to rotate and scale this basis. This is a new way of defining the process noise as far as we know. The problem itself, however, is addressed in Campbell (1990), where it is suggested to use band limited noise to avoid these problems. The idea is that the derivative of such noise exists, but the drawback is that the Kalman filter will become sub-optimal.

A system with the same structure as (3) but in discrete time will be referred to as a discrete-time descriptor system. Such systems may also be noncausal, but are easier to handle since the noncausality here means dependence on future values of the noise or the input. An application for such systems is discrete-time state-space systems with constraints. For an example see Schön et al. (2003). In the discrete-time case much work has already been done, for example on Kalman filtering see, e.g., Dai (1987), Deng and Liu (1999), Nikoukhah et al. (1998, 1999), Darouach et al. (1993), Dai (1989a). In the continuous-time case much less work has been done on statistical methods. However, some attempts to introduce white noise in the continuous case has been done as well, see, e.g., Schein and Denk (1998), Winkler (2003).

2 Derivation of the Process Noise Subspace

We will omit the deterministic input in this derivation for notational convenience, so the continuous-time linear invariant differential-algebraic equations considered has the form (4). The reader is referred to Gerdin et al. (2003) for details on how the noncausality with respect to the input signal $u(t)$ can be handled.

$$E\dot{x}(t) + Fx(t) = Bw(t), \quad (4a)$$

$$y(t) = Cx(t) + e(t). \quad (4b)$$

The E , F , and C matrices in (4) are constant matrices. For the purpose of this discussion we will assume that w and e are continuous-time white noises. (See Åström (1970) for a thorough treatment of continuous-time white noise). If $\det(Es+F)$ is not identically zero as a function of $s \in \mathbf{R}$, (4) can always be transformed into the *standard form* (6) (Brennan et al., 1996). Note that if $\det(Es+F)$ is identically zero, then $x(t)$ is not uniquely determined by $w(t)$ and the initial value $x(0)$. This can be realized by Laplace transforming (4). Therefore it is a reasonable assumption that $\det(Es+F)$ is not identically zero.

2.1 Time Domain Derivation

First, a transformation to the standard form is needed. This is done by finding a suitable change of variables $x = Qz$ and a matrix P to multiply (4a) from the left. Both P and Q are nonsingular matrices. By doing this we get

$$PEQ\dot{z}(t) + PFQz(t) = PBw(t), \quad (5)$$

which for suitably chosen P - and Q -matrices can be written in the following standard form:

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} \begin{pmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{pmatrix} + \begin{pmatrix} -A & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} z_1(t) \\ z_2(t) \end{pmatrix} = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} w(t), \quad (6)$$

where the N -matrix is *nilpotent*, i.e., $N^k = 0$ for some k . The matrices P and Q can be calculated using, e.g., ideas from Varga (1992) involving the generalized real Schur

form and the generalized Sylvester equation. We can also write (6) on the form (7) (Dai, 1989b, Ljung and Glad, 2003).

$$\dot{z}_1(t) = Az_1(t) + G_1 w(t), \quad (7a)$$

$$z_2(t) = \sum_{i=0}^{k-1} (-N)^i G_2 \frac{d^i w(t)}{dt^i}. \quad (7b)$$

From a theoretical point of view G_1 can be chosen arbitrarily, since it describes how white noise should enter an ordinary differential equation. However, constraints on G_1 can of course be imposed by the physics of the system that is modeled. When it comes to G_2 , the situation is different, here we have to find a suitable parameterization. The problem is now that white noise cannot be differentiated, so we proceed to find a condition on the B -matrix in (4a) under which there does not occur any derivatives in (7b), i.e., $N^i G_2 = 0$ for all $i \geq 1$. This is equivalent to that $NG_2 = 0$. The result is given in the following theorem.

Theorem C.1

The condition to avoid differentiation of white noise is equivalent to requiring that

$$B \in \mathcal{R}(M), \quad (8)$$

where M is a matrix derived from the standard form (6) (see the proof for details on how M is derived).

The expression $B \in \mathcal{R}(M)$ means that B is in the range of M , that is the columns of B are linear combinations of the columns of M .

Proof: Let the $n \times n$ matrix N in (6) have the singular value decomposition (SVD)

$$N = UDV^T. \quad (9)$$

Since it is nilpotent it is also singular, so m diagonal elements in D are zero. Partition $V = (V_1, V_2)$, where V_2 contains the last m columns of V having zero singular values. Then $NV_2 = 0$, and we can write $G_2 = V_2 T$, where T is an arbitrary $m \times m$ matrix, which parameterizes all matrices G_2 that satisfies $NG_2 = 0$.

According to (5) and (6) we have

$$B = P^{-1} \begin{pmatrix} G_1 \\ G_2 \end{pmatrix}. \quad (10)$$

If we now let $P^{-1} = (R_1, R_2)$, we can write (10) as

$$B = P^{-1} \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} = (R_1 \quad R_2) \begin{pmatrix} G_1 \\ V_2 T \end{pmatrix} = \underbrace{(R_1 \quad R_2 V_2)}_M \begin{pmatrix} G_1 \\ T \end{pmatrix}, \quad (11)$$

where both G_1 and T can be chosen arbitrarily. This calculation gives that

$$B \in \mathcal{R}(M), \quad (12)$$

is a condition for avoiding differentiation of the white noise signal $w(t)$. \square

The B -matrices satisfying (12) will thus allow us to incorporate white noise without having a problem with differentiation of white noise. The design parameters to be specified are G_1 and T , defined in the proof above. Also note that the requirement that white noise should not be differentiated is related to the concept of *impulse controllability* discussed in Dai (1989b).

2.2 Frequency Domain Derivation

The same condition on the noise can be derived in the frequency domain, as shown in this section. Throughout the section, we need some concepts from the theory of matrix fraction descriptions (MFD). We start by defining the *row degree* of a polynomial matrix and the concept of a *row reduced* MFD according to Rugh (1996).

Definition C.1. *The i^{th} row degree of a polynomial matrix $P(s)$, written as $r_i[P]$, is the degree of the highest degree polynomial in the i^{th} row of $P(s)$.*

Definition C.2. *If the polynomial matrix $P(s)$ is square and nonsingular, then it is called row reduced if*

$$\deg(\det P(s)) = r_1[P] + \cdots + r_n[P]. \quad (13)$$

We will use the following theorem from Kailath (1980):

Theorem C.2

If $D(s)$ is row reduced, then $D^{-1}(s)N(s)$ is proper if and only if each row of $N(s)$ has degree less than or equal the degree of the corresponding row of $D(s)$, i.e., $r_i[N] \leq r_i[D], i = 1, \dots, n$.

To utilize the results we need to write (4a) as a matrix fraction description. A MFD of (4a) is

$$X(s) = (Es + F)^{-1}BW(s). \quad (14)$$

According to Rugh (1996) a matrix fraction description can be converted to row reduced form by pre-multiplication of a unimodular¹ matrix $U(s)$. That is, $D(s)$ is row reduced in the MFD

$$X(s) = D^{-1}(s)N(s)W(s), \quad (15)$$

where $D(s) = U(s)(Es + F)$ and $N(s) = U(s)B$ for a certain unimodular matrix $U(s)$. Now, Theorem C.2 shows that the transfer function of the system is proper if the highest degree of the polynomials in each row in $N(s)$ is lower than or equal to the highest degree of the polynomials in the corresponding row of $D(s)$. This gives a condition on B in the following way:

Writing $U(s)$ as

$$U(s) = \sum_{i=0}^m U_i s^i \quad (16)$$

¹A polynomial matrix is called unimodular if its determinant is a nonzero real number (Kailath, 1980).

and writing the j^{th} row of U_i as U_{ij} , shows that the condition

$$U_{ij}B = 0 \quad i > r_j[D], \quad j = 1, \dots, n \quad (17)$$

guarantees that the transfer function of the system is proper.

Conversely, assume that (17) does not hold. Then some row degree of $N(s)$ is higher than the corresponding row degree of $D(s)$, so the transfer function is then according to Theorem C.2 not proper.

This discussion proves the following theorem.

Theorem C.3

The transfer function of the system (4) is proper if and only if

$$U_{ij}B = 0 \quad i > r_j[D], \quad j = 1, \dots, n. \quad (18)$$

Note that the criterion discussed in this section requires that the MFD is transformed to row reduced form, and an algorithm for finding this transformation is provided in Rugh (1996).

We have now proved two theorems, one using time domain methods and one using frequency domain methods, that gives conditions which are equivalent to that no white noise is differentiated in (4). This means that these two conditions are equivalent as well. The frequency domain method is good in the sense that we do not have to compute the standard form (6). However, if we want to discretize the equations it is worthwhile to compute the standard form. Once this is done the celebrated Kalman filter can be used to estimate the internal variables $x(t)$. In the subsequent section we will discuss the discretization and the estimation problems.

3 Filtering

3.1 Discretization

If the noise enters the system according to a B -matrix satisfying Theorem C.1 or C.3 the original system (4) can be written as

$$\dot{z}_1(t) = Az_1(t) + G_1w(t), \quad (19a)$$

$$z_2(t) = G_2w(t), \quad (19b)$$

$$y(t) = CQz(t) + e(t), \quad (19c)$$

where $x = Qz$. Furthermore $w(t)$ and $e(t)$ are both assumed to be Gaussian white noise signals with covariances R_1 and R_2 respectively, and zero cross-covariance (the case of nonzero cross-covariance can be handled as well, the only difference is that the expressions are more involved).

System (19) can be discretized using standard techniques from linear systems theory (Rugh, 1996). If we assume that $w(t)$ remains constant during one sample interval² (here it is assumed that sampling interval is one to simplify the notation),

$$w(t) = w[k], \quad k \leq t < (k+1), \quad (20)$$

²See, e.g., Gustafsson (2000) for a discussion on other possible assumptions on the stochastic process $w(t)$ when it comes to discretization.

we obtain

$$z_1[k+1] = \tilde{A}z_1[k] + \tilde{G}_1 w[k], \quad (21a)$$

$$z_2[k] = G_2 w[k], \quad (21b)$$

$$y[k] = CQz[k] + e[k], \quad (21c)$$

where

$$\tilde{A} = e^A, \quad \tilde{G}_1 = \int_0^1 e^{A\tau} d\tau G_1. \quad (22)$$

Hence (21) and (22) constitutes a discrete-time model of (4).

3.2 Kalman Filter

In order to apply the Kalman filter to the discrete-time model (21) we start out by rewriting (21c) as

$$\begin{aligned} y[k] &= CQz[k] + e[k] = (\tilde{C}_1 \tilde{C}_2) \begin{pmatrix} z_1[k] \\ z_2[k] \end{pmatrix} + e[k] = \tilde{C}_1 z_1[k] + \tilde{C}_2 z_2[k] + e[k] \\ &= \tilde{C}_1 z_1[k] + \underbrace{\tilde{C}_2 G_2 w[k] + e[k]}_{\tilde{e}[k]}. \end{aligned} \quad (23)$$

Combining (21a) and (23) we obtain

$$z_1[k+1] = \tilde{A}z_1[k] + \tilde{G}_1 w[k], \quad (24a)$$

$$y[k] = \tilde{C}_1 z_1[k] + \tilde{e}[k]. \quad (24b)$$

Note that the measurement noise $\tilde{e}[k]$ and the process noise $w[k]$ are correlated. Now, the Kalman filter can be applied to (24) in order to estimate the internal variables $z_1[k]$ and the process noise $w[k]$. Finally an estimate of the internal variables $z_2[k]$ can be found using the estimated process noise, since $z_2[k] = G_2 w[k]$, according to (21b). Finally the internal variables, $x[k]$, are obtained by $x[k] = Q^{-1} z[k]$. For details on the Kalman filter see Glad and Ljung (2000).

4 Example

In this example we will treat a system composed of two rotating masses as shown in Figure 1. The two rotating parts are described by the torques M_1, M_2, M_3 , and M_4 and the angular velocities z_1 and z_2 . The equations describing this system are

$$J_1 \dot{z}_1 = M_1 + M_2, \quad (25a)$$

$$J_2 \dot{z}_2 = M_3 + M_4, \quad (25b)$$

$$M_2 = -M_3, \quad (25c)$$

$$z_1 = z_2. \quad (25d)$$

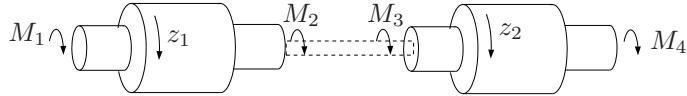


Figure 1: Two interconnected rotating masses.

Written on the form (4) these equations are

$$\begin{pmatrix} J_1 & 0 & 0 & 0 \\ 0 & J_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \dot{x} + \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \end{pmatrix} x = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} M_1 \\ M_4 \end{pmatrix}, \quad (26)$$

where $x = (z_1, z_2, M_2, M_3)^T$. Note that the matrix in front of \dot{x} is singular, hence (26) is a differential-algebraic equation. Using the following transformation matrices P and Q

$$P = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ \frac{J_2}{J_1+J_2} & -\frac{J_1}{J_1+J_2} & \frac{J_2}{J_1+J_2} & 0 \end{pmatrix}, \quad Q = \begin{pmatrix} \frac{1}{J_1+J_2} & \frac{J_2}{J_1+J_2} & 0 & 0 \\ \frac{1}{J_1+J_2} & -\frac{J_1}{J_1+J_2} & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (27)$$

the equations can be written in the standard form (6):

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{J_1 J_2}{J_1+J_2} & 0 & 0 \end{pmatrix} \dot{z} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} z = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ \frac{J_2}{J_1+J_2} & -\frac{J_1}{J_1+J_2} \end{pmatrix} \begin{pmatrix} M_1 \\ M_4 \end{pmatrix}. \quad (28)$$

Now to the important part, if we want to incorporate noise into the differential-algebraic equation (26), by adding Bw to (26), which B -matrices are allowed?

To answer this question Theorem C.1 can be consulted. We begin by calculating the matrices R_1 , R_2 and V_2 from (27) and (28). We have that

$$N = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{J_1 J_2}{J_1+J_2} & 0 & 0 \end{pmatrix} \Rightarrow V_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad (29)$$

and

$$P^{-1} = \begin{pmatrix} \frac{J_1}{J_1+J_2} & 0 & -1 & 1 \\ \frac{J_2}{J_1+J_2} & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \Rightarrow R_1 = \begin{pmatrix} \frac{J_1}{J_1+J_2} \\ \frac{J_2}{J_1+J_2} \\ 0 \\ 0 \end{pmatrix}, \quad R_2 = \begin{pmatrix} 0 & -1 & 1 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}. \quad (30)$$

We can now calculate the M matrix:

$$M = \begin{pmatrix} R_1 & R_2 V_2 \end{pmatrix} = \begin{pmatrix} \frac{J_1}{J_1+J_2} & -1 & 1 \\ \frac{J_2}{J_1+J_2} & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (31)$$

As the requirement was that $B \in \mathcal{R}(M)$ this simply means that we cannot directly add white noise to (25d) (if $J_1 > 0$ and $J_2 > 0$). This result makes physical sense, as a step change in the angular velocity would require an infinite torque.

The same condition on B can also be calculated in the frequency domain using Theorem C.3. Transforming the system to row reduced form gives that

$$U(s) = \begin{pmatrix} -\frac{1}{J_1} & -\frac{1}{J_2} & 0 & s \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \underbrace{\begin{pmatrix} -\frac{1}{J_1} & -\frac{1}{J_2} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{U_0} + \underbrace{\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{U_1} s \quad (32)$$

and that

$$D(s) = \begin{pmatrix} 0 & 0 & \frac{1}{J_1} & -\frac{1}{J_2} \\ 0 & J_2 s & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \end{pmatrix}, \quad (33)$$

with notation from section 2.2.

This gives that the row degrees of $D(s)$ are $r_1[D] = 0$, $r_2[D] = 1$, $r_3[D] = 0$, and $r_4[D] = 0$. Theorem C.3 thus gives that the transfer function is proper if and only if

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} B = 0. \quad (34)$$

What (34) says is that the last row of B must be zero, which is the same conclusion as was reached using the time domain method, Theorem C.1.

5 Discrete-Time Linear Descriptor Systems

The discrete linear time invariant descriptor system is an equation on the form

$$Ex[k+1] + Fx[k] = Bw[k], \quad (35a)$$

$$y[k] = Cx[k] + e[k], \quad (35b)$$

where E , F , and C are constant matrices and $w[k]$ and $e[k]$ are white noise sequences, i.e., sequences of independent and identically distributed random variables. For this case it is possible to make the same transformation as for a continuous differential-algebraic equation if $\det(Ez + F)$ is not identically zero as a function of $z \in \mathbf{R}$ (Section 2) since

the structure is similar. Similarly to the continuous-time case, $x[k]$ will not be uniquely determined by $w(k)$ if $\det(Ez + F)$ is identically zero. A certain transformation

$$PEQx[k+1] + PFQx[k] = PBw[k], \quad (36)$$

with nonsingular matrices P and Q will thus give us the form

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} \begin{pmatrix} z_1[k+1] \\ z_2[k+1] \end{pmatrix} + \begin{pmatrix} -A & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} z_1[k] \\ z_2[k] \end{pmatrix} = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} w[k]. \quad (37)$$

As in the continuous-time case, we can write (37) in the form

$$z_1[k+1] = Az_1[k] + G_1w[k], \quad (38a)$$

$$z_2[k] = \sum_{i=0}^{n-1} (-N)^i G_2 w[k+i]. \quad (38b)$$

The system (35) is thus well defined for all B -matrices, since no derivatives occur in this case. However, $z_2[k]$ will depend on future values of the noise. To avoid this, the noise sequence can be time shifted. If we let $\tilde{w}[k] = w[k+n-1]$ we can rewrite (38) according to

$$z_1[k+1] = Az_1[k] + G_1\tilde{w}[k-n+1], \quad (39a)$$

$$z_2[k] = \sum_{i=-n+1}^0 (-N)^i G_2 \tilde{w}[k+i], \quad (39b)$$

which can be transformed to a normal state-space description. This state-space description can then be used to implement a Kalman filter, which is discussed in Dai (1987). Other approaches to Kalman filtering of discrete-time linear descriptor systems are discussed in, Deng and Liu (1999), Nikoukhah et al. (1998, 1999), Darouach et al. (1993), Dai (1989a).

The sequences $w[k]$ and $\tilde{w}[k]$ will have the same statistical properties since they both are white noise sequences.

It can be also be noted that the same requirement that was put on B in the continuous-time case may also be used in the discrete-time case. This would then guarantee that the system would not depend on future noise values and the noise sequence would not have to be time shifted.

5.1 Frequency Domain

The ideas of time shifting the noise might become clearer if they are treated in the frequency domain. If we transform (35) to the frequency domain we get

$$X(z) = \underbrace{(Ez + F)^{-1} B}_{H(z)} W(z). \quad (40)$$

The only difference from a transfer function for a state-space system is that here $H(z)$ is noncausal in the general case. If we rewrite (40) as

$$X(z) = \underbrace{H(z)z^{-T}}_{\tilde{H}(z)} \underbrace{z^T W(z)}_{\tilde{W}(z)}, \quad (41)$$

then $\tilde{W}(z)$ will be a time shifted version of $W(z)$ and $\tilde{H}(z)$ will be a causal transfer function if T is large enough.

6 Conclusion

We have in this paper proposed a framework for modeling and filtering of systems composed of linear differential-algebraic equations. The main reason for studying these systems is that they occur as the natural description delivered from object-oriented modeling software. At the core of this problem we find the question of how to incorporate stochastics into linear differential-algebraic equations. This has been solved in this paper in the case where white noise is used. The result was presented as two equivalent theorems, one in the time domain and one in the frequency domain. The resulting model fits into the optimal filtering framework and standard methods such as the Kalman filter applies. An example was also given, which showed that the conditions derived for how the noise can enter the system gives requirements which are physically motivated.

7 Acknowledgment

The authors would like to thank the Swedish Research Council and the Foundation for Strategic Research for supporting this work.

References

- Åström, K. J. (1970). *Introduction to Stochastic Control Theory*. Mathematics in Science and Engineering. Academic Press, New York and London.
- Brenan, K. E., Campbell, S. L., and Petzold, L. R. (1996). *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM's Classics in Applied Mathematics. SIAM, New York, USA.
- Campbell, S. L. (1990). Descriptor systems in the 90's. In *Proceedings of the 29th Conference on Decision and control*, pages 442–447, Honolulu, Hawaii, USA.
- Dai, L. (1987). State estimation schemes for singular systems. In *Proceedings of the 10th IFAC World Congress*, volume 9, pages 211–215, Munich, Germany.
- Dai, L. (1989a). Filtering and LQG problems for discrete-time stochastic singular systems. *IEEE Transactions on Automatic Control*, 34(10):1105–1108.
- Dai, L. (1989b). *Singular Control Systems*. Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, Germany.
- Darouach, M., Zasadzinski, M., and Mehdi, D. (1993). State estimation of stochastic singular linear systems. *International Journal of Systems Sciences*, 24(2):345–354.
- Deng, Z. L. and Liu, Y. M. (1999). Descriptor Kalman filtering. *International Journal of Systems Science*, 30(11):1205–1212.

- Gerdin, M., Glad, T., and Ljung, L. (2003). Parameter estimation in linear differential-algebraic equations. In *Proceedings of the 13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands.
- Glad, T. and Ljung, L. (2000). *Control Theory, Multivariable and Nonlinear Methods*. Taylor and Francis.
- Gustafsson, F. (2000). *Adaptive Filtering and Change Detection*. John Wiley & Sons, New York, USA.
- Kailath, T. (1980). *Linear systems*. Prentice Hall, Englewood Cliffs, NJ, USA.
- Ljung, L. and Glad, T. (2003). *Modellbygge och simulering*. Studentlitteratur. In Swedish.
- Mattsson, S. E., Elmquist, H., and Otter, M. (1998). Physical system modeling with Modelica. *Control Engineering Practice*, 6:501–510.
- Nikoukhah, R., Campbell, S. L., and Delebecque, F. (1998). Kalman filtering for general discrete-time linear systems. In *Proceedings of the 37th Conference on Decision and Control*, pages 2886–2891, Tampa, FL, USA.
- Nikoukhah, R., Campbell, S. L., and Delebecque, F. (1999). Kalman filtering for general discrete-time linear systems. *IEEE Transactions on Automatic Control*, 44(10):1829–1839.
- Rugh, W. J. (1996). *Linear System Theory*. Information and system sciences series. Prentice Hall, Upper Saddle River, NJ, USA, second edition.
- Schein, O. and Denk, G. (1998). Numerical solution of stochastic differential-algebraic equations with applications to transient noise simulation of microelectronic circuits. *Journal of Computational and Applied Mathematics*, 100:77–92.
- Schön, T., Gustafsson, F., and Hansson, A. (2003). A note on state estimation as a convex optimization problem. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 61–64, Hong Kong.
- Tiller, M. (2001). *Introduction to Physical Modeling with Modelica*. Kluwer Academic Press, Boston, MA, USA.
- Varga, A. (1992). Numerical algorithms and software tools for analysis and modelling of descriptor systems. In *Proceedings of the 2nd IFAC Workshop on System Structure and Control*, Prague, Czechoslovakia, pages 392–395.
- Winkler, R. (2003). Stochastic differential algebraic equations of index 1 and applications in circuit simulation. *Journal of Computational and Applied Mathematics*, pages 477–505.

Paper D

A Note on State Estimation as a Convex Optimization Problem

Authors: Thomas Schön, Fredrik Gustafsson, and Anders Hansson.

Edited version of paper originally published in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 61–64, Hong Kong, April 2003.

Preliminary version published as Technical Report LiTH-ISY-R-2497, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

A Note on State Estimation as a Convex Optimization Problem

Thomas Schön, Fredrik Gustafsson, and Anders Hansson

Department of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden.
E-mail: (schon, fredrik, hansson)@isy.liu.se

Abstract

The Kalman filter computes the maximum a posteriori (MAP) estimate of the states for linear state-space models with Gaussian noise. We interpret the Kalman filter as the solution to a convex optimization problem, and show that we can generalize the MAP state estimator to any noise with log-concave density function and any combination of linear equality and convex inequality constraints on the states. We illustrate the principle on a hidden Markov model, where the state vector contains probabilities that are positive and sum to one.

Keywords: State estimation, Kalman filter, convex optimization, hidden Markov models, linear regression.

1 Introduction

STATE estimation in stochastic linear models is an important problem in many model based approaches in signal processing and automatic control applications, where the Kalman filter is the standard method. However, if we have prior information of some kind it is often impossible to incorporate this in the Kalman filter framework. We will in this paper show how we can use prior information by considering the optimization problem that the Kalman filter solves. A similar treatment can be found in Robertson and Lee (2002), however they only consider quadratic problems, whereas we will consider a larger class of convex problems.

2 Convex Optimization

In this section we will give a very brief introduction to convex optimization, see also Boyd and Vandenberghe (2001).

The main message in convex optimization is that one should *not* differ between linear and nonlinear optimization problems, but instead between convex and non-convex problems. This is due to the fact that the class of convex problems is much larger than that covered by linear problems, and we know that for a convex problem any local optimum is also a global optimum. Moreover, there exist efficient algorithms for solving convex optimization problems. A convex optimization problem is defined as

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 0, \dots, m, \\ & a_i^T x = b_i, \quad i = 0, \dots, n, \end{aligned} \quad (1)$$

where the functions f_0, \dots, f_m are convex and the equality constraints are linear. We will in the following sections try to identify some estimation problems that can be cast as convex optimization problems.

3 Notation and Background

Maximum a posteriori (MAP) estimation (Jazwinski, 1970) is about finding an estimator of a stochastic variable z that maximizes the conditional density $p(z|y)$, given the observation y ($y \in \mathbf{R}^{n_y}$ and $z \in \mathbf{R}^{n_z}$). Thus, the MAP problem is

$$\max_z \log(p(z|y)). \quad (2)$$

In the sequel, the measurement vectors y_i from time 0 to time t will be denoted $y_{0:t}$, and similarly $z_{0:t}$ denotes all unknowns including the initial values. The operator $z_i^{(j)}$ extracts the j th element from the vector z_i .

The assumptions commonly used in the literature are that the elements in the z vectors are spatially and temporally independent (white noise) and Gaussian distributed. We will insist on the independence assumption, but not on the assumption of Gaussian densities, giving us the following form of $\log(p(z))$ (suppressing the dependence on y)

$$\log(p(z_{0:t})) = \log\left(\prod_{i=0}^t p_{z_i}(z_i)\right) = \sum_{i=0}^t \log(p_{z_i}(z_i)). \quad (3)$$

Depending on the distribution, the objective function in (1) can be explicitly written as in Table 1, see also Boyd and Vandenberghe (2001).

4 Convex Optimization Estimation

In this section we will discuss the estimation problem in the presence of constraints. In Table 1 the objective functions are given for several log-concave¹ densities. Constraints arise in the derivation of some of these probability density functions (PDF), but constraints

¹A function $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is *log-concave* if $f(x) > 0$ for all x in the domain of f , and $\log(f)$ is a concave function (Boyd and Vandenberghe, 2001).

Table 1: Objective functions in (1) for different normalized (zero mean and unit covariance) probability density functions.

PDF	Objective function	Extra constraints
Gaussian	$\sum_{i=0}^t \ z_i\ ^2$	
Exponential	$\sum_{i=0}^t \sum_{j=1}^{n_z} z_i^{(j)} - 1$	$z \geq 0$
Laplacian	$\sum_{i=0}^t \sum_{j=1}^{n_z} z_i^{(j)} $	
Uniform	constant	$-\sqrt{3} \leq z \leq \sqrt{3}$

also arise from prior information of some kind, e.g., a model assumption. This will be discussed in Section 6.

Assume we want to estimate $(x^T, z^T)^T$, where z has a certain known distribution, and that x and z are related through the constraints

$$A \begin{pmatrix} x \\ z \end{pmatrix} = b, \quad (4)$$

If we now want to use (2) we are faced with the problem of finding the joint distribution of x and z , which can be quite tedious.

Problem 1 (Convex optimization estimation)

Assume that $p(z)$ is a known log-concave probability density function. Then, the MAP-estimate for $(x^T, z^T)^T$, where x and z are related via (4) is given by

$$\begin{aligned} & \max_{x,z} \log(p_z(z)) \\ & \text{s.t.} \quad A \begin{pmatrix} x \\ z \end{pmatrix} = b. \end{aligned} \quad (5)$$

Remark: Any linear equalities and convex inequalities may be added to this formulation, and standard software applies.

This approach to estimation is presented in Boyd and Vandenberghe (2001). The standard estimation problem is to interpret x as the parameters conditioned on the measurements $x|y$, and then z is just a nuisance parameter. The standard approach, not often written explicitly, is to marginalize the nuisance parameters to get

$$p(x|y) = \int p(x|y, z)p(z|y) dz, \quad (6)$$

where the constraints are used explicitly. This works fine in a range of applications, and the solution most often has a quite simple form. In the general case, we can formulate the problem below.

5 Linear Regression Example

As an example of estimation, consider a linear regression problem in matrix form

$$Y = \Phi^T \theta + E. \quad (7)$$

Interpret $E \leftrightarrow z$ as a Gaussian nuisance parameter with variance σ^2 , the regression parameter $\theta \leftrightarrow x$ as the parameter and $Y, \Phi \leftrightarrow y$ as the observations. The well-known result from marginalization is that

$$\theta \in \mathcal{N}((\Phi\Phi^T)^{-1}\Phi Y, \sigma^2(\Phi\Phi^T)^{-1}). \quad (8)$$

Alternatively, we can pose the problem as

$$\begin{aligned} \max_{\theta, E} \quad & \log(p_E(E)) \\ \text{s.t.} \quad & (\Phi^T \quad \mathbf{1}) \begin{pmatrix} \theta \\ E \end{pmatrix} = Y. \end{aligned} \quad (9)$$

If this regression model happens to be an ARX model of a transfer function

$$G(e^{i\omega}) = \frac{\sum_l b^{(l)} e^{-i\omega l}}{1 + \sum_l a^{(l)} e^{-i\omega l}}, \quad (10)$$

in system identification, we use $\theta = (a^T, b^T)^T$. Now, we can simply add constraints such as bounded DC gain $L \leq G(0) \leq U$, or more generally, any lower and upper bound on the transfer function

$$L(\omega) \leq \frac{\sum_l b^{(l)} e^{-i\omega l}}{1 + \sum_l a^{(l)} e^{-i\omega l}} \leq U(\omega), \quad (11)$$

which is easily rewritten in the standard form. Similarly, any other interval for any other frequency of the transfer function can be bounded.

6 Convex Optimization Filtering

In Section 4 we talked about constraints in general. We will in this section discuss a special type of constraints, namely the ones that appear in describing the dynamic behavior of a model. In order to obtain convex problems we will use linear models of the dynamics. The following model

$$Ex_{t+1} = Ax_t + Bw_t + Ke_t, \quad (12a)$$

$$y_t = Cx_t + De_t, \quad (12b)$$

together with a density for the initial state p_{x_0} and $p_{e_t}(e_t)$, $p_{w_t}(w_t)$ will constitute our model. With $E = I$, $K = 0$ we have the standard state-space model, and with $E = I$, $B = 0$, $D = I$ we have the so called *innovation form*. If the E -matrix in (12a) is invertible we can rewrite the equation in a state-space model. Otherwise we have what is commonly referred to as a *descriptor model* (Luenberger, 1977).

To put state filtering in the general estimation form as in Problem 1, let

$$z = (x_0^T \quad w_{0:t-1}^T \quad e_{0:t}^T)^T, \quad (13)$$

and interpret x as $x_{1:t}|y_{1:t}$. To rewrite the conditional density more explicitly, use the independence assumption and (3), which gives

$$\log(p(x_0, w_{0:t-1}, e_{0:t})) = \log(p_{x_0}(x_0)) + \sum_{i=0}^{t-1} \log(p_{w_i}(w_i)) + \sum_{i=0}^t \log(p_{e_i}(e_i)). \quad (14)$$

Using Bayes' theorem, $p(z|y) = p(y|z)p(z)/p(y)$ and the fact that

$$p(x_t) = p_{x_0}(x_0) \prod_{i=0}^{t-1} p_{w_i}(w_i), \quad (15a)$$

$$p(y_t|x_t) = \prod_{i=0}^t p_{e_i}(e_i), \quad (15b)$$

we obtain the following objective function

$$p(x_0, w_{0:t-1}, e_{0:t}) = \prod_{i=0}^t p_{e_i}(e_i) p_{x_0}(x_0) \prod_{i=0}^{t-1} p_{w_i}(w_i). \quad (16)$$

Conditioned on z in (13), the states in (12) are uniquely defined by a deterministic mapping $x = f(z)$, which implies that $p(x|z) = f(z)$ contains nothing stochastic. That is, the MAP-estimate of x and z are simply related by $\hat{x}^{MAP} = f(\hat{z}^{MAP})$. Similarly, the joint MAP-estimate x, z in the convex optimization formulation is given by maximizing $p(z)$, since $p(z, x) = p(x|z)p(z) = f(z)p(z)$ by Bayes' theorem. Hence, we have now justified the following general convex filtering problem.

Problem 2 (Convex optimization filtering)

Assume that the probability density functions $p_{x_0}(x_0)$, $p_{w_i}(w_i)$, and $p_{e_i}(e_i)$ are log-concave. In the presence of constraints in terms of a dynamic model (12) the MAP-estimate is the solution $\hat{x}_t = x_t$ to the following problem

$$\begin{aligned} \max_{x_{0:t}, z} \quad & \log(p_{x_0}(x_0)) + \sum_{i=0}^{t-1} \log(p_{w_i}(w_i)) + \sum_{i=0}^t \log(p_{e_i}(e_i)) \\ \text{s.t.} \quad & E_i x_{i+1} = A_i x_i + B_i w_i + K_i e_i, \quad i = 0, \dots, t-1, \\ & y_i = C_i x_i + D_i e_i, \quad i = 0, \dots, t. \end{aligned}$$

Remark: Any linear equalities and convex inequalities may be added to this formulation, and standard software applies.

As is evident from Problem 2 we see that we are free to use different densities for the different disturbances $p_{x_0}(x_0)$, $p_{w_i}(w_i)$, and $p_{e_i}(e_i)$. It is here also worth noting that the recursive solution to Problem 2 under the assumptions of Gaussian densities and a nonsingular E -matrix is the celebrated Kalman filter. This has been known for a long time, see, e.g., Sorenson (1970), Kailath (1974) for nice historical accounts of this fact, and for a proof see Rao (2000). It is also worthwhile noting that Problem 2 under the assumption of Gaussian disturbances is a weighted least-squares problem. To see this

combine 2 and the Gaussian case in Table 1, where the weights are the inverse of the covariance matrices. This provides a deterministic interpretation of the problem that the Kalman filter solves. For more on the similarities and differences between deterministic and stochastic filtering, see, e.g., Kailath et al. (2000). We also see that if we solve Problem 2 we will not only obtain the filtered estimate $\hat{x}_{t|t}$, but also all the smoothed estimates, $\hat{x}_{i|t}$, $i = 0, \dots, t - 1$.

So why should we solve the estimation problem via 2, which demands more computations, instead of via the Kalman filter? There are two reasons. The first reason is that we can handle all log-concave density functions, not just the Gaussian. The second reason is that we can add any prior information, in convex form, to problem 2. That is we can add linear equality constraints and convex inequality constraints, and still find the optimal estimate. We will see an illustration of this in the example in the subsequent section.

7 HMM Example

There are mainly two filtering problems, where there exist finite-dimensional recursive optimal filters, and in particular a finite-dimensional MAP-estimator. One is, as already mentioned, linear state-space models with Gaussian noise. Here the Kalman filter is optimal in ML, MAP and minimum variance senses. For non-Gaussian noises, the Kalman filter computes the linear state estimate with minimum variance, but it is no longer the MAP or ML estimator.

The other case is *hidden Markov models* (HMM). Interestingly, it has been pointed out (Andersson, 2002) that the HMM can be written as a state-space model. That is, the Kalman filter computes the best possible linear estimate of the Markov state. This fact makes it possible to compare conceptually different approaches on the same example!

A hidden Markov model is defined by a discrete variable $\xi \in (1, 2, \dots, n)$ with a known transition probability matrix A , where $A^{(i,j)} = \Pr(\xi_t = i | \xi_{t-1} = j)$, that is, given that $\xi = j$ at time $t - 1$, the probability that $\xi = i$ at time t is $A^{(i,j)}$. We will assume an observation process $\nu \in (1, 2, \dots, m)$, where $\Pr(\nu = i | \xi = j) = C^{(i,j)}$. The filter for computing the a posteriori probabilities can be expressed as the recursion

$$\pi_t^{(i)} = p(\xi_t = i) = \frac{\sum_{j=1}^n \sum \pi_{t-1}^{(j)} A^{(j,i)} C^{(\nu_t, j)}}{\sum_{j=1}^n \sum \pi_{t-1}^{(j)} C^{(\nu_t, j)}}. \quad (17a)$$

The MAP-estimate is $\hat{\xi}_t = \arg \max_i \pi_t^{(i)}$. Now, the HMM can be written as the state-space model

$$x_{t+1} = Ax_t + w_t, \quad (18a)$$

$$y_t = Cx_t + e_t, \quad (18b)$$

where $x_t^{(i)} = \Pr(\xi_t = i)$ and $y_t^{(i)} = \Pr(\nu_t = i)$. This is the state-space form (12) with $B = D = E = I$, $K = 0$) where the disturbances are zero-mean white noises, and the stationary covariance matrices can be shown to be

$$Q = \text{Cov}\{w_t\} = \text{diag}(\pi) - A \text{diag}(\pi) A^T, \quad (19a)$$

$$R = \text{Cov}\{e_t\} = \text{diag}(C\pi) - C \text{diag}(\pi) C^T, \quad (19b)$$

where π is the stationary solution to (in vector form)

$$\pi = \lim_{t \rightarrow \infty} A^t \pi_0, \quad \text{where} \quad \pi_0 > 0. \quad (20)$$

Since the states x we are estimating in a HMM are probabilities we have the following prior information on the states

$$\sum_{i=1}^2 x^{(i)} = 1, \quad \text{and} \quad x^{(i)} \geq 0, \quad i = 1, 2. \quad (21)$$

In the standard Kalman filter it is impossible to incorporate this prior information about the states, however in Problem 2 it is straightforward. We will now examine four different filters using an increasing amount of prior information (in 1-3 we have approximated w_t and e_t in (18) as Gaussian with zero-mean and covariances (19)):

1. The Kalman filter.
2. The convex optimization filter with constraint $\sum_i x_t^{(i)} = 1$. This case can alternatively be computed by the Kalman filter using $P_0 = p_0 \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ and any $\sum_i x_0^{(i)} = 1$, or by using the fictitious measurement $y_0 = (1, 1, \dots, 1)x_0 = 1$ with zero measurement noise. Note, however, that the Riccati equation will be singular here, which may imply certain numerical difficulties. A more theoretically sound alternative is given in Andersson (2002).
3. The convex optimization filter with constraint (21).
4. The optimal filter (17).

Table 2: RMSE values for the different filters.

1. Kalman filter	0.585
2. 2 with $x_1 + x_2 = 1$	0.573
3. 2 with $x_1 + x_2 = 1$ and $x \geq 0$	0.566
4. Optimal filter	0.403

The numerical example is taken from Andersson (2002), where

$$A = C = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}. \quad (22)$$

In Table 2, the root mean square error (RMSE) is given for these four cases and in Figure 1 the states are shown. From this table it is obvious that we can obtain better estimates by using more information in this case. Of course, the convex optimization filters cannot compare to the performance of the optimal filter. However, the point is to show the flexibility of the approach, and the problem of consideration can be generalized with more constraints or a more complicated measurement relation, such that the optimal filter does no longer exist.

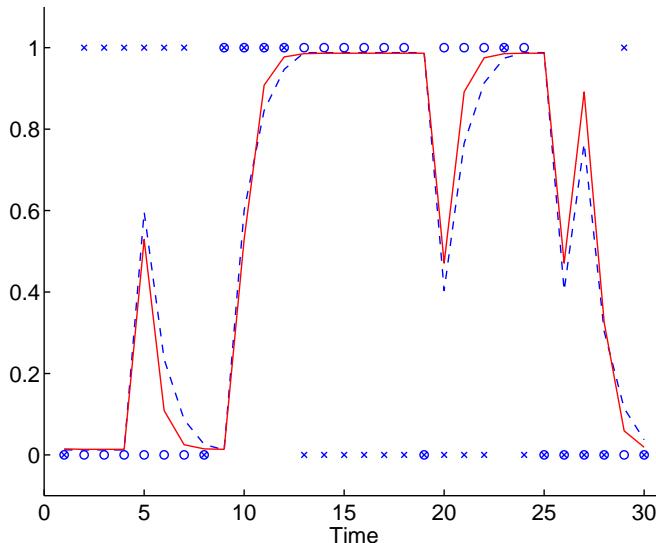


Figure 1: The true state is marked by \circ , and the measured states by \times . The dashed/solid line is the estimate from filter 3, respective 4.

8 Conclusion

We have formulated the state estimation problem in a convex optimization framework. In this way, well-known numerical efficient algorithms can be used to compute the MAP-estimate of the state vector, without any problems with local minima. Compared to the Kalman filter, the advantage is that any log-concave noise densities can be used and any linear equality or convex inequality constraints may be included, while the main drawback is that no recursive convex optimization algorithm is yet available, which makes the approach computer intensive.

References

- Andersson, S. (2002). *Hidden Markov Models - Traffic Modeling and Subspace Methods*. PhD thesis, Centre for Mathematical Sciences, Mathematical Statistics, Lund University, Lund, Sweden.
- Boyd, S. and Vandenberghe, L. (2001). Convex optimization.
- Jazwinski, A. H. (1970). *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York, USA.
- Kailath, T. (1974). A view of three decades of linear filtering theory. *IEEE Transactions on Information Theory*, IT-20(2):146–181.

- Kailath, T., Sayed, A. H., and Hassibi, B. (2000). *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, NJ, USA.
- Luenberger, D. G. (1977). Dynamic equations in descriptor form. *IEEE Transactions on Automatic Control*, AC-22(3).
- Rao, C. V. (2000). *Moving Horizon Strategies for the Constrained Monitoring and Control of Nonlinear Discrete-Time Systems*. PhD thesis, University of Wisconsin Madison.
- Robertson, D. G. and Lee, J. H. (2002). On the use of constraints in least squares estimation and control. *Automatica*, 38:1113–1123.
- Sorenson, H. W. (1970). Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum*, 7:63–68.

Paper E

Particle Filters for System Identification of State-Space Models Linear in Either Parameters or States

Authors: Thomas Schön and Fredrik Gustafsson.

Edited version of paper originally published in *Proceedings of the 13th IFAC Symposium on System Identification*, pages 1287–1292, Rotterdam, The Netherlands, September 2003.

Preliminary version published as Technical Report LiTH-ISY-R-2518, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Particle Filters for System Identification of State-Space Models Linear in Either Parameters or States

Thomas Schön and Fredrik Gustafsson

Department of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden.
E-mail: (schon, fredrik)@isy.liu.se.

Abstract

The potential use of the marginalized particle filter for nonlinear system identification is investigated. The particle filter itself offers a general tool for estimating unknown parameters in nonlinear models of moderate complexity, and the basic trick is to model the parameters as a random walk (so called roughening noise) with decaying variance. We derive algorithms for systems which are linear in either the parameters or the states, but generally not in both. In these cases, marginalization applies to the linear part, which firstly significantly widens the scope of the particle filter to more complex systems, and secondly decreases the variance in the linear parameters/states for fixed filter complexity. This second property is illustrated in an example of a chaotic model. The particular case of freely parameterized linear state-space models, common in subspace identification approaches, is bilinear in states and parameters, and thus both cases above are satisfied. One can then choose which one to marginalize.

Keywords: System identification, nonlinear estimation, recursive estimation, particle filters, Kalman filters, Bayesian estimation, marginalization, Rao-Blackwellization.

1 Introduction

IN this contribution, the particle filter (Gordon et al., 1993, Doucet et al., 2001a) is applied to some classical system identification problems (Ljung, 1999) based on time-varying parametric state-space models

$$z_{t+1} = f_t(z_t, \theta) + w_t^z, \quad (1a)$$

$$y_t = h_t(z_t, \theta) + e_t, \quad (1b)$$

where $z \in \mathbf{R}^{n_z}$ is the state variable, $\theta \in \mathbf{R}^{n_\theta}$ is the parameter vector, and $y \in \mathbf{R}^{n_y}$ is the output variable. The additive noise terms are assumed to be independent and identically distributed (i.i.d.).

First, we briefly review the problem formulation given in the accompanying paper by Gustafsson and Hriljac (2003). By augmenting the state vector with the parameters, $x_t = (z_t^T, \theta_t^T)^T$, and assuming a random walk parameter variation (of which constant parameters is a special case), we get

$$\begin{pmatrix} z_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} f_t(z_t, \theta_t) \\ \theta_t \end{pmatrix} + \begin{pmatrix} w_t^z + v_t^z \\ w_t^\theta + v_t^\theta \end{pmatrix}, \quad (2a)$$

$$y_t = h_t(z_t, \theta_t) + e_t, \quad (2b)$$

where the noises are physical state noise w_t^z , state roughening noise v_t^z , parameter random walk for time-varying parameters w_t^θ and parameter roughening noise v_t^θ . The roughening noise is instrumental in the particle filter to get good performance, and is a second level design parameter. For system identification, $w_t^\theta = 0$ and v_t^θ has a variance decaying to zero, which yields converging parameter estimates. The particle filter recursively approximates the filter density function $p(x_t|Y_t)$, where $Y_t \triangleq \{y_i\}_{i=0}^t$, and the approximation converges to the true filter density when the number of particles tends to infinity. The only problem is that the practical limit for “infinity” depends on the dimension of x_t , that is, the sum of number of parameters θ_t and states, z_t .

Now, if there is linear substructure available in the model this can be exploited using *marginalization*. Conceptually, marginalization means that the linear states are marginalized out. We can then apply optimal filters for the linear states and the particle filter is only applied to the truly nonlinear states. In this way, the samples in the particle filter will live in a lower dimensional space. Hence, we will intuitively obtain *more accurate estimates* for a given number of samples, since we use the optimal filters for a part of the state vector. Alternatively, we can apply the particle filter on *more complex models*. These are the practical implications of our contribution.

We will in this contribution consider the two following special cases of (1a):

1. The model is affine in the parameters and possibly nonlinear in the states,

$$f(z_t, \theta_t) = f_t^z(z_t) + A_t(z_t)\theta_t, \quad (3a)$$

$$h(z_t, \theta_t) = h_t(z_t) + C_t(z_t)\theta_t. \quad (3b)$$

2. The model is affine in the states and possibly nonlinear in the parameters,

$$f(z_t, \theta_t) = f_t^\theta(\theta_t) + A_t(\theta_t)z_t, \quad (4a)$$

$$h(z_t, \theta_t) = h_t(\theta_t) + C_t(\theta_t)z_t. \quad (4b)$$

In the subsequent two sections we will introduce the particle filter and the marginalization technique used for variance reduction. In Section 4 the models under consideration are introduced and we discuss the connections to subspace identification. Section 5 is devoted to applying the marginalized particle filter to nonlinear system identification problem posed by a chaotic system. Finally, the conclusions are given in Section 6.

2 Particle Filter

We here briefly present the theory and main algorithm. For a more intuitive presentation, see the accompanying paper, Gustafsson and Hriljac (2003).

2.1 Recursive Bayesian Estimation

Consider systems that are described by the generic state-space model (2). The optimal Bayesian filter in this case is given below. For further details, consult Doucet et al. (2001a), Jazwinski (1970).

Denote the observations at time t by $Y_t \triangleq \{y_i\}_{i=0}^t$. The Bayesian solution to compute the filter density, $p(x_t|Y_t)$, of the state vector, given past observations, is given by Jazwinski (1970)

$$p(x_{t+1}|Y_t) = \int p(x_{t+1}|x_t)p(x_t|Y_t) dx_t, \quad (5a)$$

$$p(x_t|Y_t) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})}. \quad (5b)$$

For expressions on $p(x_{t+1}|x_t)$ and $p(y_t|x_t)$ in (5) we use the known probability densities $p_{e_t}(x)$ and $p_{v_t+w_t}(x)$, with all noises assumed independent,

$$p(x_{t+1}|x_t) = p_{v_t+w_t}(x_{t+1} - f(x_t)), \quad (6a)$$

$$p(y_t|x_t) = p_{e_t}(y_t - h(x_t)). \quad (6b)$$

2.2 Implementation

A numerical approximation to (5) is given by

$$p(x_t|Y_t) \approx \sum_{i=1}^N \tilde{q}_t^{(i)} \delta\left(x_t - x_t^{(i)}\right), \quad (7)$$

where $\delta(\cdot)$ is Dirac's delta function. The particles $x_t^{(i)}$ and the corresponding weights $\tilde{q}_t^{(i)}$ represent a sampled version of the filter density $p(x_t|Y_t)$, and intuitively, the more samples the better approximation (Doucet et al., 2001a).

2.3 Algorithm

The discussion in the previous section is summarized in the algorithm below. This is the algorithm presented by Gordon et al. (1993) under the name, *Bayesian bootstrap filter*. The particle filter can be interpreted as a simulation-based method, i.e., N possible

state trajectories $\{x_t^{(i)}\}_{i=1}^N$ are simulated. Based on the measurements each trajectory is assigned a weight $\tilde{q}_t^{(i)}$ representing the probability of that trajectory being the correct one.

Algorithm E.1 (Particle filter)

1. *Initialization:* For $i = 1, \dots, N$, initialize the particles, $x_{0|-1}^{(i)} \sim p_{x_0}(x_0)$.
 2. *Measurement update:* For $i = 1, \dots, N$, evaluate the importance weights $q_t^{(i)} = p(y_t | x_{t|t-1}^{(i)})$ and normalize $\tilde{q}_t^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^N q_t^{(j)}}$.
 3. *Resample with replacement* N particles according to
$$\Pr\left(x_{t|t}^{(i)} = x_{t|t-1}^{(j)}\right) = \tilde{q}_t^{(j)}. \quad (8)$$
 4. *Time update:* For $i = 1, \dots, N$, predict new particles according to
$$x_{t+1|t}^{(i)} \sim p\left(x_{t+1|t} | x_{t|t}^{(i)}\right). \quad (9)$$
 5. Set $t := t + 1$ and iterate from step 2.
-

3 Marginalization for Variance Reduction

Consider the case where the model is linear in some of the states. Then the Kalman filter can be used to estimate the linear states, denoted x_t^l , and the particle filter can be used to estimate the nonlinear states, denoted x_t^n . To separate the problem of estimating $p(x_t^l, x_t^n | Y_t)$ into one linear and one nonlinear problem, Bayes' theorem is used

$$p(x_t^l, X_t^n | Y_t) = p(x_t^l | X_t^n, Y_t)p(X_t^n | Y_t). \quad (10)$$

Here the density $p(x_t^l | X_t^n, Y_t)$ is given by the Kalman filter and the particle filter is used to estimate $p(X_t^n | Y_t)$. This means that the particles live in a lower-dimensional space, and it can indeed be proven (Doucet et al., 2001b, Nordlund, 2002) that the variance of any function of the state and parameter is decreased or remains constant when using marginalization for a given number of particles. This technique of marginalizing out the linear state is also referred to as Rao-Blackwellization (Doucet et al., 2001b).

Let the entity we want to estimate for some inference function $g(\cdot)$ be given by

$$I(g(x_t)) = E_{p(x_t | Y_t)}\{g(x_t)\} = \int g(x_t)p(x_t | Y_t)dx_t. \quad (11)$$

Furthermore, let the estimate of (11) using N particles and the standard particle filter be denoted by $\hat{I}_N^s(g(x_t))$. When the marginalized particle filter is used the same estimate is denoted by $\hat{I}_N^m(g(x_t))$. Then there is a central limit theorem stating that for large N we have

$$\hat{I}_N^s(g(x_t)) \approx \mathcal{N}(x_t | I(g(x_t)), R_s(N)), \quad (12a)$$

$$\hat{I}_N^m(g(x_t)) \approx \mathcal{N}(x_t | I(g(x_t)), R_m(N)), \quad (12b)$$

where

$$R_s(N) \geq R_m(N). \quad (13)$$

For details concerning this result, see, e.g., Doucet et al. (1999, 2001b), Nordlund (2002).

Asymptotically as the number of particles tend to infinity there is nothing to gain in using marginalization, since then the particle filter will provide a perfect description of $p(x_t^l, x_t^n | Y_t)$. However, since we only can use a finite number of particles it is certainly useful to marginalize and use the optimal filter, i.e., the Kalman filter, for the linear states. For details concerning the marginalized particle filter, see, e.g., Chen and Liu (2000), Doucet et al. (2001b), Nordlund (2002).

4 Models

In this section it will be shown how the particle filter can be used to estimate the nonlinear states and the Kalman filter to estimate the linear states, using the marginalization technique discussed above. All noise terms associated with the linear states are here assumed to be Gaussian, which means that the optimal estimator for the linear states/parameters is given by the Kalman filter. For the details concerning the Kalman filter equations, the state transition densities, and the likelihood functions in Algorithms E.2 and E.3 the reader is referred to Nordlund (2002). First there will be a discussion on models that are linear in the states and nonlinear in the parameters. This is followed by the reversed case, i.e., linear in the parameters and nonlinear in the states.

4.1 State-Space Models Linear in the States

A state-space model linear in the states and possibly nonlinear in the parameters is written as

$$z_{t+1} = f_t^z(\theta_t) + A_t(\theta_t)z_t + w_t^z, \quad (14a)$$

$$\theta_{t+1} = \theta_t + v_t^\theta, \quad (14b)$$

$$y_t = h_t(\theta_t) + C_t(\theta_t)z_t + e_t, \quad (14c)$$

where $v_t^\theta \sim \mathcal{N}(0, Q_t^{v,\theta})$ and $w_t^z \sim \mathcal{N}(0, Q_t^{w,z})$ ¹. Note that we can let the roughening noise v_t^z be zero when using marginalization. The filter density will here be separated using Bayes' theorem according to

$$p(z_t, \Theta_t | Y_t) = p(z_t | \Theta_t, Y_t)p(\Theta_t | Y_t). \quad (15)$$

Note that we here consider the parameter trajectory Θ_t , but only the last state vector z_t . The first density on the right hand side in (15) is given by the Kalman filter, while the second one is approximated by the particle filter. That is, we randomize particles in the parameter space according to our prior, and then each particle trajectory will be associated with one Kalman filter. The exact algorithm is given below.

¹The noise on the nonlinear part, here v_t^θ , can in fact have an arbitrary distribution. Similarly, The pdf $p_{\theta_0}(\theta_0)$ does not have any restrictions, since it is only used in the particle filter, the same goes for $p_{e_t}(e_t)$ if $C = 0$ in (14c). However, we leave these generalizations as a remark and assume Gaussian distributions.

Algorithm E.2 (Particle filter for linear states)

1. *Initialization:* For $i = 1, \dots, N$, initialize the particles, $\theta_{0|-1}^{(i)} \sim p_{\theta_0}(\theta_0)$ and set $\{z_{0|-1}^{(i)}, P_{0|-1}^{(i)}\} = \{\bar{z}_0, \bar{P}_0\}$.
2. *Particle filter measurement update:* Let $C_t^{(i)} = C_t(\theta_{t|t-1}^{(i)})$ and $h_t^{(i)} = h(\theta_{t|t-1}^{(i)})$. For $i = 1, \dots, N$, evaluate the importance weights

$$q_t^{(i)} = p(y_t | \Theta_t^{(i)}, Y_{t-1}) = \mathcal{N}\left(y_t | h_t + C_t^{(i)} z_{t|t-1}^{(i)}, C_t^{(i)} P_{t|t-1}^{(i)} (C_t^{(i)})^T + R_t\right),$$
and normalize $\tilde{q}^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^N q_t^{(j)}}$.
3. *Resample with replacement* N particles according to, $\Pr\left(\theta_{t|t}^{(i)} = \theta_{t|t-1}^{(j)}\right) = \tilde{q}_t^{(j)}$.
4. *Particle filter time update and Kalman filter*

- (a) *Kalman filter measurement update:* Let $h_t^{(i)} = h_t(\theta_{t|t}^{(i)})$, $C_t^{(i)} = C_t(\theta_{t|t}^{(i)})$.

$$z_{t|t}^{(i)} = z_{t|t-1}^{(i)} + K_t^{(i)} \left(y_t - h_t^{(i)} - C_t^{(i)} z_{t|t-1}^{(i)} \right), \quad (16a)$$

$$P_{t|t}^{(i)} = P_{t|t-1}^{(i)} - K_t^{(i)} M_t^{(i)} \left(K_t^{(i)} \right)^T, \quad (16b)$$

$$M_t^{(i)} = C_t^{(i)} P_{t|t-1}^{(i)} \left(C_t^{(i)} \right)^T + R_t, \quad (16c)$$

$$K_t^{(i)} = P_{t|t-1}^{(i)} \left(C_t^{(i)} \right)^T \left(M_t^{(i)} \right)^{-1}. \quad (16d)$$

- (b) *Particle filter time update:* For $i = 1, \dots, N$, predict new particles,

$$\theta_{t+1|t}^{(i)} \sim p\left(\theta_{t+1|t} | \Theta_t^{(i)}, Y_t\right) = \mathcal{N}\left(\theta_{t+1} | \theta_{t|t}^{(i)}, Q_t^{v,\theta}\right).$$

- (c) *Kalman filter time update:* Let $A_t^{(i)} = A_t(\theta_{t+1|t}^{(i)})$ and $f_t^{z,(i)} = f_t^z(\theta_{t+1|t}^{(i)})$.

$$z_{t+1|t}^{(i)} = A_t^{(i)} z_{t|t}^{(i)} + f_t^{z,(i)}, \quad (17a)$$

$$P_{t+1|t}^{(i)} = A_t^{(i)} P_{t|t}^{(i)} \left(A_t^{(i)} \right)^T + Q_t^{w,z}. \quad (17b)$$

5. Set $t := t + 1$ and iterate from step 2.
-

Comparing the algorithms E.1 and E.2 we see that the differences are in the prediction step, which now consists of a Kalman filter update stage (split into step 4(a) and 4(c)) besides the prediction of the nonlinear states.

In some cases the same Riccati recursion can be used for all the particles, and hence a lot of computations can be saved. This occurs when the matrices A_t and C_t in (14) are independent of θ_t . In this case $P_{t|t}^{(i)} = P_{t|t}$ for all $i = 1, \dots, N$ and hence the covariance only has to be updated once for each t . More on this can be found in Gustafsson et al. (2002).

4.2 State-Space Models Linear in the Parameters

A state-space model that is linear in the parameters can be written as

$$z_{t+1} = f_t^z(z_t) + A_t(z_t)\theta_t + w_t^z, \quad (18a)$$

$$\theta_{t+1} = \theta_t + v_t^\theta, \quad (18b)$$

$$y_t = h_t^z(z_t) + C_t(z_t)\theta_t + e_t. \quad (18c)$$

In this case the filter density will be split the other way around, compared to the previous section, i.e.,

$$p(Z_t, \theta_t | Y_t) = p(\theta_t | Z_t, Y_t)p(Z_t | Y_t). \quad (19)$$

The last density is approximated by the particle filter, while the first one can be solved by a Kalman filter for a parameter estimation problem in a linear regression framework. The corresponding algorithm will thus be

Algorithm E.3 (Particle filter for linear parameters)

1. Initialization: For $i = 1, \dots, N$, initialize the particles, $z_{0|1}^{(i)} \sim p_{z_0}(z_0)$ and set $\{\theta_{0|1}^{(i)}, P_{0|1}^{(i)}\} = \{\bar{\theta}_0, \bar{P}_0\}$.
2. Particle filter measurement update: Let $h_t^{(i)} = h_t(z_{t|t-1}^{(i)})$, $C_t^{(i)} = C_t(z_{t|t-1}^{(i)})$. For $i = 1, \dots, N$, evaluate the importance weights
 $q_t^{(i)} = p(y_t | Z_t^{(i)}, Y_{t-1}) = \mathcal{N}\left(y_t | h_t^{(i)} + C_t^{(i)}\theta_{t|t-1}, C_t^{(i)}P_{t|t-1}^{(i)}(C_t^{(i)})^T + R_t\right)$,
and normalize $\tilde{q}_t^{(i)} = q_t^{(i)} / \sum_{j=1}^N q_t^{(j)}$.
3. Resample with replacement N particles according to, $\Pr\left(z_{t|t}^{(i)} = z_{t|t-1}^{(j)}\right) = \tilde{q}_t^{(j)}$.
4. Particle filter time update and Kalman filter

- (a) Kalman filter measurement update: Let $h_t^{(i)} = h_t(z_{t|t}^{(i)})$, $C_t^{(i)} = C_t(z_{t|t}^{(i)})$.

$$\theta_{t|t}^{(i)} = \theta_{t|t-1}^{(i)} + K_t^{(i)} \left(y_t - h_t^{(i)} - C_t^{(i)}\theta_{t|t-1}^{(i)} \right), \quad (20a)$$

$$P_{t|t}^{(i)} = P_{t|t-1}^{(i)} - K_t^{(i)} M_t^{(i)} \left(K_t^{(i)} \right)^T, \quad (20b)$$

$$M_t^{(i)} = C_t^{(i)} P_{t|t-1}^{(i)} \left(C_t^{(i)} \right)^T + R_t, \quad (20c)$$

$$K_t^{(i)} = P_{t|t-1}^{(i)} \left(C_t^{(i)} \right)^T \left(M_t^{(i)} \right)^{-1}. \quad (20d)$$

- (b) Particle filter time update: Let $f_t^{z,(i)} = f_t^z(z_{t+1|t}^{(i)})$ and $A_t^{(i)} = A_t(z_{t+1|t}^{(i)})$. For $i = 1, \dots, N$, predict new particles,

$$\begin{aligned} z_{t+1|t}^{(i)} &\sim p\left(z_{t+1|t} | Z_t^{(i)}, Y_t\right) \\ &= \mathcal{N}\left(z_{t+1} | f_t^{z,(i)} + A_t^{(i)}\theta_{t|t}^{(i)}, A_t^{(i)}P_{t|t}^{(i)}(A_t^{(i)})^T + Q_t^{w,z}\right). \end{aligned}$$

(c) Kalman filter time update: Let $f_t^{z,(i)} = f_t^z(z_{t+1|t}^{(i)})$ and $A_t^{(i)} = A_t(z_{t+1|t}^{(i)})$.

$$\theta_{t+1|t}^{(i)} = \theta_{t|t}^{(i)} + L_t^{(i)} \left(z_{t+1|t}^{(i)} - f_t^{z,(i)} - A_t^{(i)} \theta_{t|t}^{(i)} \right), \quad (21a)$$

$$P_{t+1|t}^{(i)} = P_{t|t}^{(i)} + Q_t^{v,\theta} - L_t^{(i)} N_t^{(i)} \left(L_t^{(i)} \right)^T, \quad (21b)$$

$$N_t^{(i)} = A_t^{(i)} P_{t|t}^{(i)} \left(A_t^{(i)} \right)^T + Q_t^{w,z}, \quad (21c)$$

$$L_t^{(i)} = P_{t|t}^{(i)} \left(A_t^{(i)} \right)^T \left(N_t^{(i)} \right)^{-1}. \quad (21d)$$

5. Set $t := t + 1$ and iterate from step 2.

The measurements used in the Kalman filter are thus the “normal” measurements y_t and the predicted state trajectory $z_{t+1|t}$, i.e., the samples from the particle filter. Step 4(c) in the current algorithm contains a measurement update, using the prediction (since this contains information about θ_t) from the particle filter, and a time update. An interesting special case of the two different model types discussed above is when we consider “the intersection” of the two types, i.e., a model that is bilinear in the states z_t and in the parameters θ_t .

A particular case of interest is a general state-space model in innovation form

$$z_{t+1} = A(\theta_t)z_t + K(\theta_t)e_t, \quad (22a)$$

$$y_t = C(\theta_t)z_t + e_t, \quad (22b)$$

where the parameters enter linearly in A , K , and C . The filter density will here be split according to (19). One popular approach here is so called subspace identification (Van Overschee and De Moor, 1996). This class of algorithms usually perform very well and provides consistent estimates. One limitation is that it is hard to give the density function for the parameters, even in the Gaussian case, and this is perhaps where the particle filter can help. This case is mentioned to show the relation to classical system identification problems.

Assume, to avoid ambiguities in the state coordinates, an observer canonical form and scalar output, where $C = (1, 0, \dots, 0)$ and that all parameters in A and K are unknown. Then, given the state trajectory and measurement, we have from (22) the linear regression $z_{t+1} = Az_t + K(y_t - (1, 0, \dots, 0)z_t)$. This regression problem has to be solved for each particle $z_t^{(i)}$, $i = 1, \dots, N$.

In the case where there are more states to be estimated than parameters, i.e., $\dim z_t > \dim \theta$ it is better to split the density $p(Z_t, \theta_t | Y_t)$ in (19) the other way around, i.e., as in (15). This time, a Kalman filter estimating the states z_t for each particle $\theta_t^{(i)}$ is needed. In this way the dimension of the state estimated by the particle filter is kept as low as possible. An example where this situation typically occurs is in gray box identification (Ljung, 1999).

5 Chaos Example

The ideas presented in this paper will be illustrated using the following chaotic model

$$z_{t+1} = (1 - z_t)z_t\theta + w_t, \quad (23a)$$

$$y_t = z_t + e_t, \quad (23b)$$

where z_t is the state variable, y_t is the measurement, θ is the unknown parameter, w_t is the process noise, and e_t is the measurement noise. Both these noise densities are Gaussian distributed. The aim is to recursively estimate both the state z_t and the parameter θ . This model is linear in the time-invariant parameter θ and nonlinear in the state z_t . This fits our framework, according to Section 4.2 and hence Algorithm E.3 can be applied. This problem has also been studied in Gustafsson and Hriljac (2003), where the particle filter was directly applied to the augmented state $x_t = (z_t, \theta_t)^T$. Model (23) can be written in the form (18), i.e.,

$$z_{t+1} = A_t(z_t)\theta_t + w_t^z + v_t^z, \quad (24a)$$

$$\theta_{t+1} = \theta_t + v_t^\theta, \quad (24b)$$

$$y_t = h_t(z_t) + e_t, \quad (24c)$$

where $A_t(z_t) = (1 - z_t)z_t$ and $h_t(z_t) = z_t$. The two noises $v_t^z \sim \mathcal{N}(0, Q_t^{v,z})$ and $v_t^\theta \sim \mathcal{N}(0, Q_t^{v,\theta})$ are roughening noises. Furthermore, $e_t \sim \mathcal{N}(0, R_t)$.

In the simulations, two different particle filters were used, the standard particle filter, Algorithm E.1, applied to the augmented state vector x_t and the marginalized particle filter according to Algorithm E.3. The true value of θ is 3.92, and the initial guess is $\theta_{0|-1} \sim \mathcal{N}(3.83, 0.04)$. The initial state is $z_0 \sim \mathcal{N}(0, 1)$. We do not use any process noise, however we have roughening noises $Q_0^{v,z} = Q_0^{v,\theta} = 10^{-2}$, which is decreased at each time step, according to Gustafsson and Hriljac (2003). The measurement noise has variance $R_t = 10^{-5}$, and we have used 200 Monte Carlo simulations. In Figure 1 the filtered estimates of θ are shown using these two algorithms for 150, 1000, and 10000 particles respectively. In order to make the difference more apparent the Root Mean Square Error (RMSE) is plotted in Figure 2 as a function of the number of particles used in the simulations. Note that the RMSE values are calculated from time 50. In that way the transient effects are not included in the RMSE values. According to (13) the estimates should be better or the same when we use the marginalized particle filter. From Figure 2 we can see that this is indeed the case. It is only the estimate of the linear part θ that is improved, this is also consistent with the theory, see, e.g., Nordlund (2002) for the theoretical details. That this is true in the simulations is apparent by Figure 2, from which it is clear that the estimate of the linear part (top) clearly is better using the marginalized particle filter. The estimate of the nonlinear part z_t has the same quality. If we could use an infinite number of particles the results using the standard and the marginalized particle filter would have been the same, since then the particle filter would be able to provide an arbitrarily good estimate of $p(x_t|Y_t)$. We can see indications of this fact in the top plot in Figure 2, since the more particles that are used the closer the estimates get to each other.

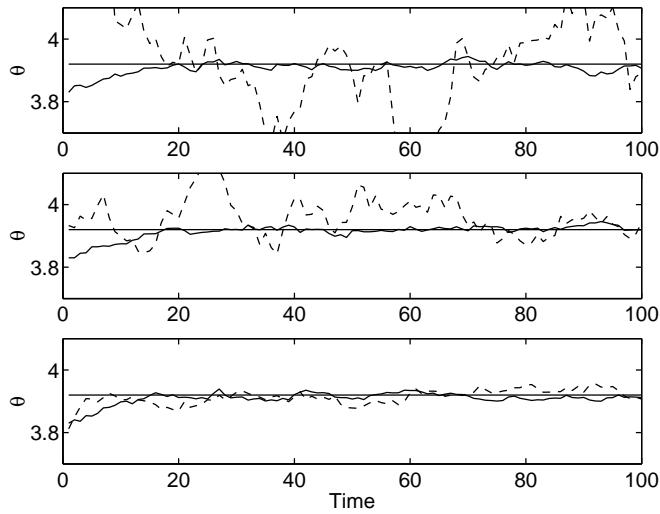


Figure 1: Estimates of θ using the standard (dashed) and the marginalized (solid) particle filters. The true θ is shown using a solid line. Top plot – 150 particles, middle – 1000 particles, bottom – 10000 particles.

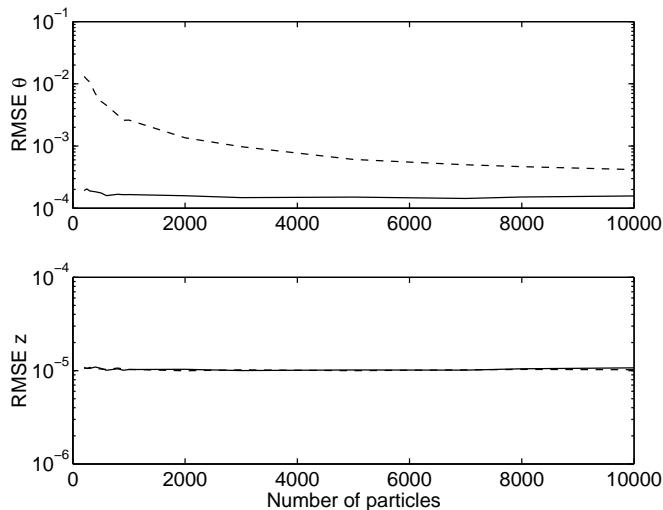


Figure 2: RMSE values for $\hat{\theta}$ (top) and \hat{z}_t (bottom) as a function of the number of particles used. Notice that a log-scale has been used in the plots, and that a dashed line has been used for the standard particle filter and a solid line for the marginalized particle filter.

6 Conclusion

The potential use of particle filtering for identification of unknown parameters in non-linear systems was explained in the accompanying paper, Gustafsson and Hriljac (2003). Here, we have proposed the use of marginalized particle filters. More specifically, we studied the cases where the model is either linear in the states and nonlinear in the parameters, or nonlinear in the states and linear in the parameters. The algorithms were given for these two cases. It is straightforward to give the algorithm for an arbitrary mix of linear and nonlinear states and parameters. The advantage of marginalization is that one can apply the filter to larger problems with more states and parameters, or that fewer particles and thus less filter complexity is needed for a given performance. Finally an example was given, which illustrates the improvement in estimation performance compared to using a standard particle filter.

References

- Chen, R. and Liu, J. S. (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society, 62(3)*:493–508.
- Doucet, A., de Freitas, N., and Gordon, N., editors (2001a). *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, USA.
- Doucet, A., Gordon, N., and Krishnamurthy, V. (1999). Particle filters for state estimation of jump Markov linear systems. Technical Report CUED/F-INFENG/TR 359, Signal Processing Group, Department of Engineering, University of Cambridge, Trupington street, CB2 1PZ Cambridge.
- Doucet, A., Gordon, N., and Krishnamurthy, V. (2001b). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing, 49(3)*:613–624.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113.
- Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J. (2002). Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing, 50(2)*:425–437.
- Gustafsson, F. and Hriljac, P. (2003). Particle filters for prediction of chaos. In *Proceedings of the 13th IFAC Symposium on System Identification*, pages 1281–1286, Rotterdam, The Netherlands.
- Jazwinski, A. H. (1970). *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York, USA.
- Ljung, L. (1999). *System identification, Theory for the user*. System sciences series. Prentice Hall, Upper Saddle River, NJ, USA, second edition.

Nordlund, P.-J. (2002). *Sequential Monte Carlo Filters and Integrated Navigation*. Licentiate Thesis No 945, Department of Electrical Engineering, Linköping University, Sweden.

Van Overschee, P. and De Moor, B. (1996). *Subspace identification for linear systems - theory, implementation, applications*. Kluwer Academic Publishers.

Paper F

Maximum Likelihood Nonlinear System Estimation

Authors: Thomas B. Schön, Adrian Wills, and Brett Ninness.

Edited version of paper accepted for publication in *Proceedings of the 14th IFAC Symposium on System Identification*, Newcastle, Australia, March 2006.

Preliminary version published as Technical Report LiTH-ISY-R-2713, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Maximum Likelihood Nonlinear System Estimation

Thomas B. Schön*, Adrian Wills†, and Brett Ninness†

*Department of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden.
E-mail: schon@isy.liu.se.

†School of Electrical Engineering and Computer Science,
University of Newcastle,
Callaghan, NSW 2308, Australia.
E-mail: (Adrian.Wills, Brett.Ninness)@newcastle.edu.au.

Abstract

This paper is concerned with the parameter estimation of a relatively general class of nonlinear dynamic systems. A Maximum Likelihood (ML) framework is employed in the interests of statistical efficiency, and it is illustrated how an Expectation Maximization (EM) algorithm may be used to compute these ML estimates. An essential ingredient is the employment of so-called particle smoothing methods to compute required conditional expectations via a Monte Carlo approach. A simulation example demonstrates the efficacy of these techniques.

Keywords: Nonlinear systems, system identification, maximum likelihood, expectation maximisation algorithm, particle smoother.

1 Introduction

THE significance but difficulty of estimating parameterizations of nonlinear system classes is widely recognised (Ljung, 2003, Ljung and Vicino, 2005). This has led to approaches that focus on specific system classes such as those described by Volterra kernel (Bendat, 1990), neural network (Narendra and Parthasarathy, 1990), nonlinear AR-MAX (NARMAX) (Leontaritis and Billings, 1985), and Hammerstein – Wiener (Rangan et al., 1995) structures.

The paper here considers the estimation of a certain class of nonlinear systems that can be represented in state-space form whereby state and measurement noise enter additively and the parameter dependence is affine. To estimate this nonlinear model structure parameterization, a Maximum Likelihood (ML) criterion will be employed, principally in recognition of the general statistical efficiency of such an approach. Of course, the use of an ML approach (for example, with regard to linear dynamic systems) is common, and it is customary to employ a gradient-based search technique such as a damped Gauss – Newton method to actually compute the estimates (Ljung, 1999, Söderström and Stoica, 1989). This requires the computation of a cost Jacobian which typically necessitates implementing one filter, derived (in the linear case) from a Kalman filter, for each parameter that is to be estimated.

An alternative, recently explored in Gibson et al. (2005) in the context of bilinear systems is to employ the expectation maximisation algorithm (Dempster et al., 1977) for the computation of ML estimates. Unlike gradient-based search, which is applicable to maximisation of any differentiable cost function, EM methods are only applicable to maximisation of likelihood functions. However, the dividend of this specialization is that they do not require computation of gradients, and are well recognised as being particularly robust against attraction to local minima (Gibson and Ninness, 2005).

Given these recommendations, this paper develops and demonstrates an EM-based approach to nonlinear system identification. This will require the computation of smoothed state estimates that, in the linear case, could be found by standard linear smoothing methods (Gibson et al., 2005). In the fairly general nonlinear context considered in this work we propose a particle-based approach whereby approximations of the required smoothed state estimates are approximated by Monte Carlo based empirical averages (Doucet et al., 2001).

It is important to acknowledge that there has been previous work related to this approach. In Andrieu et al. (2004), the possibility of incorporating the parameters into the state vector and employing particle *filtering* methods was discussed, but dismissed as untenable. Balancing this, the contributions Kitagawa (1998), Schön and Gustafsson (2003) provide evidence to question this conclusion.

Additionally, the work Doucet and Tadić (2003), Andrieu et al. (2004) has considered employing particle filters to compute the Jacobians necessary for a gradient-based approach. Finally, the contribution Andrieu et al. (2004) has also considered using the EM algorithm in conjunction with particle-based methods. However, by employing improved particle smoothing methods and by more careful numerical implementation of a key “maximisation” step, the present work is able to report significantly improved performance.

2 Problem Formulation

This paper is concerned with the following model class, which is affinely parameterized in the (unknown) parameter $\vartheta \in \mathbf{R}^{n_\vartheta}$:

$$\underbrace{\begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix}}_{z_t} = \underbrace{\begin{pmatrix} f_1(x_t, u_t, t) \\ h_1(x_t, u_t, t) \end{pmatrix}}_{\alpha_t} \vartheta + \underbrace{\begin{pmatrix} f_2(x_t, u_t, t) \\ h_2(x_t, u_t, t) \end{pmatrix}}_{\beta_t} + \underbrace{\begin{pmatrix} w_t \\ e_t \end{pmatrix}}_{\eta_t}. \quad (1)$$

Here f_1, f_2, h_1 and h_2 are arbitrary (possibly time-varying) nonlinear functions, $x_t \in \mathbf{R}^n$ is the underlying system state, $u_t \in \mathbf{R}^m$, $y_t \in \mathbf{R}^p$ are respectively (observed) multi-dimensional inputs and outputs. The initial state x_1 and noise terms w_t and e_t are assumed to be realizations from Gaussian stochastic processes given by,

$$x_1 \sim \mathcal{N}(\mu, P_1), \quad \eta_t \sim \mathcal{N}(0, \Pi). \quad (2)$$

In light of this, the model structure (1) is completely described by the parameter vector θ defined as

$$\theta^T \triangleq \begin{pmatrix} \vartheta^T & \text{vec}\{\Pi\}^T & \text{vec}\{P_1\}^T & \mu^T \end{pmatrix}. \quad (3)$$

With regard to this model structure, this paper will be solely concerned with a parameter estimate $\hat{\theta}$ of θ derived via the ML criterion

$$\hat{\theta}(Y_N) = \arg \max_{\theta} p_{\theta}(Y_N), \quad (4)$$

where $Y_N \triangleq \{y_1, \dots, y_N\}$ is an N point record of observed measurements and $p_{\theta}(Y_N)$ is then the joint probability density function of Y_N implied by the model structure (1) and a parameter value θ .

In the linear, time invariant and Gaussian case, a (possibly steady state) Kalman Filter can be used to compute this cost (and required Jacobians for gradient-based search). Here, algorithms are developed to extend this principle to the more general nonlinear model class (1). In doing so, it is recognized that, especially in the nonlinear case, it is generally hard to compute (4) since it may well represent a non-convex optimization problem. To address this issue, a central contribution of this work is the employment of the EM algorithm.

3 Expectation Maximization Algorithm

The Expectation Maximization (EM) algorithm introduced in Dempster et al. (1977) presents a non gradient-based approach for iteratively obtaining maximum likelihood estimates (4). Within areas of applied statistics, it is widely recognized for its robustness. The key idea underlying it is the consideration of an extension to (4); viz.

$$\hat{\theta}(X_N, Y_N) = \arg \max_{\theta} p_{\theta}(X_N, Y_N). \quad (5)$$

Here, an extra data set X_N , commonly referred to as the *incomplete data* or the *missing data* has been introduced. Its choice is an essential design variable, which if possible should be made so that the solution of (5) is straightforward.

The link between the two problems (4) and (5) is provided by the definition of conditional probability which implies

$$\log p_{\theta}(Y_N) = \log p_{\theta}(X_N, Y_N) - \log p_{\theta}(X_N | Y_N). \quad (6)$$

Taking expectations of both sides of this equation which are conditional on the observations Y_N and with respect to underlying density specified by θ being set at a value $\theta = \theta'$ will leave the left hand side unaltered, and hence deliver

$$L(\theta) = \underbrace{\mathbb{E}_{\theta'} \{ \log p_{\theta}(X_N, Y_N) | Y_N \}}_{\mathcal{Q}(\theta, \theta')} - \underbrace{\mathbb{E}_{\theta'} \{ \log p_{\theta}(X_N | Y_N) | Y_N \}}_{\mathcal{V}(\theta, \theta')}. \quad (7)$$

Since the logarithm is concave, Jensen's inequality establishes that $\mathcal{V}(\theta, \theta') \leq \mathcal{V}(\theta', \theta')$ and therefore choosing θ that satisfies $\mathcal{Q}(\theta, \theta') \geq \mathcal{Q}(\theta', \theta')$ implies that $L(\theta) \geq L(\theta')$. That is, values of θ that increase $\mathcal{Q}(\theta, \theta')$ beyond its value at θ' also increase the underlying log-likelihood function of interest. This implies the Expectation Maximization (EM) algorithm.

Algorithm F.1 (Expectation maximization (EM))

Given an initial estimate θ_0 , iterate the following until convergence.

E: $\mathcal{Q}(\theta, \theta_k) = \mathbb{E}_{\theta_k} \{ \log p_{\theta}(X_N, Y_N) | Y_N \}$

M: $\theta_{k+1} = \arg \max_{\theta} \mathcal{Q}(\theta, \theta_k)$

4 EM for Parameter Estimation

In agreement with previous applications of EM for parameter estimation (see discussion in Gibson et al. (2005)) we define the missing data X_N to equal the state sequence $X_N \triangleq \{x_1, \dots, x_{N+1}\}$. With this choice in place, the next step in applying the EM algorithm involves computation of $\mathcal{Q}(\theta, \theta_k)$ which may be achieved via the following Lemma.

Lemma F.1

With regard to system (1) and the above choice for missing data X_N , the function \mathcal{Q} can be expressed as

$$-2\mathcal{Q}(\theta, \theta_k) = N \log \det \Pi + \text{Tr} (\Pi^{-1} \Phi(\vartheta)) + \log \det P_1 + \text{Tr} (P_1^{-1} \Psi(\mu)) + c, \quad (8)$$

where c is a constant and with $l_t \triangleq z_t - \beta_t$,

$$\Psi(\mu) \triangleq \mathbb{E}_{\theta_k} \{ (x_1 - \mu)(x_1 - \mu)^T | Y_N \}, \quad (9a)$$

$$\Phi(\vartheta) \triangleq \sum_{t=1}^N \mathbb{E}_{\theta_k} \{ (l_t - \alpha_t \vartheta)(l_t - \alpha_t \vartheta)^T | Y_N \}. \quad (9b)$$

An essential point is that both Φ and Ψ require the computation of expectations conditional on Y_N . In the case of linear systems this can be achieved by employing a linear smoother (often called a Kalman Smoother). In the nonlinear case considered in this paper, this approach is not suitable, and alternate means for computing smoothed state estimates are required. This topic is addressed in Section 5.

In the meantime, supposing that it is possible to compute these expectations, then the second step of the EM algorithm involves maximization of \mathcal{Q} with respect to θ , which is the subject of the following Lemma.

Lemma F.2

The function $\mathcal{Q}(\theta, \theta_k)$ is maximized over θ by making the following choices

$$\vartheta = \Sigma^{-1}\Gamma, \quad (10a)$$

$$\mu = E_{\theta_k} \{x_1|Y_N\}, \quad (10b)$$

$$\Pi = \Phi(\Sigma^{-1}\Gamma), \quad (10c)$$

$$P_1 = \Psi(E_{\theta_k} \{x_1|Y_N\}), \quad (10d)$$

where as before $l_t \triangleq z_t - \beta_t$ and

$$\Sigma \triangleq \sum_{t=1}^N E_{\theta_k} \{\alpha_t^T \alpha_t | Y_N\}, \quad (10e)$$

$$\Gamma \triangleq \sum_{t=1}^N E_{\theta_k} \{\alpha_t^T l_t | Y_N\}. \quad (10f)$$

With these definitions in place, the EM algorithm for parameter estimation can be expressed in more detail as follows.

Algorithm F.2 (Expectation maximization for parameter estimation)

Given an initial parameter vector θ_0 , iterate the following steps until convergence is achieved.

1. Calculate Σ, Γ and $E_{\theta_k} \{x_1|Y_N\}$ then ϑ_k and μ_k .
 2. Calculate $\Phi(\vartheta_k)$ and $\Psi(\mu_k)$ then Π_k and P_{1_k} .
-

5 Monte Carlo Based Smoothing

In this section we examine numerical solutions of nonlinear smoothing problems that employ recursive Monte Carlo techniques. In relation to this, it is worth noting that while very significant effort has been directed towards nonlinear filtering via this sort of approach (particle filters), very little has been done when it comes to solving the nonlinear smoothing problem. See, e.g., Godsill et al. (2004), Kitagawa (1996), Tanizaki (2001) for some work in this direction.

After careful evaluation, this paper will employ the methods developed in Tanizaki (2001), where the key distinguishing idea relative to the other work mentioned above is the consideration of propagating approximations of $p(x_{t+1}, x_t|Y_N)$ rather than $p(x_t|Y_N)$. In order to explain the ideas, the paper begins by addressing the general problem of random number generation with respect to a given, possibly complicated distribution.

5.1 Random Sampling

Consider the problem of generating random numbers distributed according to some target density $t(x)$ which potentially is rather complex. One way of doing this would be to employ an alternate density that is simple to draw from, say $s(x)$, referred to as the *sampling*

density, and then calculate the probability that the sample was in fact generated from the target density. That is, a sample $x^{(i)} \sim s(x)$ is drawn, and then the following ratio is calculated

$$a(x^{(i)}) \propto \frac{t(x^{(i)})}{s(x^{(i)})}, \quad (11)$$

which indicates how probable it is that $x^{(i)}$ is in fact generated from the target density $t(x)$.

The probability of accepting $x^{(i)}$ as a sample from $t(x)$ is referred to as the *acceptance probability*, and typically it is computed via consideration of $a(x^{(i)})$. This is the case, for example, for all of the so-called “acceptance – rejection sampling”, “importance sampling/resampling” and “Metropolis – Hastings independence sampling” methods (Tanizaki, 2001). Here, as will be detailed presently, importance resampling will be employed.

5.2 Monte Carlo Based Filtering

In the case of filtering, the target density referred to above becomes $t(x_t) = p(x_t|Y_t)$, and it is then necessary to also choose an appropriate sampling density $s(\cdot)$ and acceptance probability. This is in fact quite simple, since from Bayes’ theorem and the Markov property

$$p(x_t|Y_t) = p(x_t|y_t, Y_{t-1}) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \propto p(y_t|x_t)p(x_t|Y_{t-1}), \quad (12)$$

which suggests, since $t(x) \propto a(x)s(x)$, the following choices

$$\underbrace{p(x_t|Y_t)}_{t(x_t)} \propto \underbrace{p(y_t|x_t)}_{a(x_t)} \underbrace{p(x_t|Y_{t-1})}_{s(x_t)}. \quad (13)$$

Via the principle of importance resampling the acceptance probabilities, $\{\tilde{a}^{(i)}\}_{i=1}^M$, are calculated according to

$$\tilde{a}^{(i)} = \frac{a(x_{t|t-1}^{(i)})}{\sum_{j=1}^M a(x_{t|t-1}^{(j)})} = \frac{p(y_t|x_{t|t-1}^{(i)})}{\sum_{j=1}^M p(y_t|x_{t|t-1}^{(j)})}, \quad (14)$$

where $x_{t|t-1}^{(i)} \sim p(x_t|Y_{t-1})$. That is, the acceptance probabilities $\tilde{a}^{(i)}$ depend upon computation of $p(y_t|x_{t|t-1}^{(i)})$. Via the assumption of additive noise e_t , the model (1) makes this straightforward to obtain.

The algorithm then proceeds by obtaining samples from $p(x_t|Y_t)$ by resampling the particles $\{x_{t|t-1}^{(i)}\}_{i=1}^M$ from the sampling density, $p(x_t|Y_{t-1})$, according to the corresponding acceptance probabilities, $\{\tilde{a}^{(i)}\}_{i=1}^M$. If this procedure is recursively repeated over time the following approximation

$$p(x_t|Y_t) \approx \sum_{i=1}^M \frac{1}{M} \delta(x_t - x_{t|t}^{(i)}) \quad (15)$$

is obtained, and we have in fact derived the *particle filter* algorithm, which is given in Algorithm F.3. It was first introduced in Gordon et al. (1993).

Algorithm F.3 (Particle filter)

1. Initialize the particles, $\{x_{0|t-1}^{(i)}\}_{i=1}^M \sim p_{x_0}(x_0)$. Set $t := 0$.

2. Calculate importance weights $\{q_t^{(i)}\}_{i=1}^M$ according to

$$q_t^{(i)} = p(y_t | x_{t|t-1}^{(i)}) \quad (16)$$

and normalize $\tilde{q}_t^{(i)} = q_t^{(i)} / \sum_{j=1}^M q_t^{(j)}$.

3. Resample N particles, with replacement, according to

$$\Pr(x_{t|t}^{(i)} = x_{t|t-1}^{(j)}) = \tilde{q}_t^{(j)}. \quad (17)$$

4. For $i = 1, \dots, M$, predict new particles according to

$$x_{t+1|t}^{(i)} \sim p(x_{t+1|t} | x_{t|t}^{(i)}). \quad (18)$$

5. Set $t := t + 1$ and iterate from step 2.

5.3 Particle Smoother

In solving the smoothing problem the target density becomes

$$t(x_{t+1}, x_t) = p(x_{t+1}, x_t | Y_N). \quad (19)$$

Similarly to what was discussed in the previous section we have to find a suitable sampling density and the corresponding acceptance probabilities to solve the smoothing problem. Again, using Bayes' theorem we have

$$p(x_{t+1}, x_t | Y_N) = p(x_t | x_{t+1}, Y_t) p(x_{t+1} | Y_N), \quad (20)$$

where

$$\begin{aligned} p(x_t | x_{t+1}, Y_N) &= p(x_t | x_{t+1}, Y_t, Y_{t+1:N}) = \frac{p(Y_{t+1:N} | x_t, x_{t+1}, Y_t) p(x_t | x_{t+1}, Y_t)}{p(Y_{t+1:N} | x_{t+1}, Y_t)} \\ &= p(x_t | x_{t+1}, Y_t) = \frac{p(x_{t+1} | x_t) p(x_t | Y_t)}{p(x_{t+1} | Y_t)}. \end{aligned} \quad (21)$$

Inserting (21) into (20) gives

$$\underbrace{p(x_{t+1}, x_t | Y_N)}_{t(x_{t+1}, x_t)} = \underbrace{\frac{p(x_{t+1} | x_t)}{p(x_{t+1} | Y_t)}}_{a(x_{t+1}, x_t)} \underbrace{p(x_t | Y_t) p(x_{t+1} | Y_N)}_{s(x_{t+1}, x_t)}. \quad (22)$$

At time t the sampling density can be used to generate samples. In order to find the acceptance probabilities $\{a^{(i)}\}_{i=1}^M$ we have to calculate

$$a(x_{t+1}, x_t) = \frac{p(x_{t+1}|x_t)}{p(x_{t+1}|Y_t)}, \quad (23)$$

where $p(x_{t+1}|x_t)$ is calculated using the model (1), and $p(x_{t+1}|Y_t)$ can be approximated according to

$$p(x_{t+1}|Y_t) = \int p(x_{t+1}|x_t)p(x_t|Y_t)dx_t \approx \sum_{j=1}^M \frac{1}{M} p\left(x_{t+1}|x_{t|t}^{(j)}\right),$$

where (15) has been used. The particles can now be resampled according to the normalized acceptance probabilities $\{\tilde{a}^{(i)}\}_{i=1}^M$ to generate samples from $p(x_{t+1}, x_t|Y_N)$. The above discussion can be summarized in the following algorithm (first introduced in Tanizaki (2001)),

Algorithm F.4 (Particle smoother)

1. Run the particle filter (Algorithm F.3) and store the filtered particles, $\{x_{t|t}^{(i)}\}_{i=1}^M$, $t = 1, \dots, N$. Set $t := N$.
2. Initialize the smoothed particles and importance weights at time N according to $\{x_{N|N}^{(i)} = x_{N|N}^{(i)}, \tilde{q}_{N|N}^{(i)} = 1/M\}_{i=1}^M$ and set $t := t - 1$.
3. Calculate weights $\{q_{t|N}^{(i)}\}_{i=1}^M$ according to

$$q_{t|N}^{(i)} = \frac{p\left(x_{t+1|N}^{(i)}|x_{t|t}^{(i)}\right)}{\sum_{j=1}^M p\left(x_{t+1|N}^{(j)}|x_{t|t}^{(j)}\right)} \quad (24)$$

and normalize $\tilde{q}_{t|N}^{(i)} = q_{t|N}^{(i)} / \sum_{j=1}^M q_{t|N}^{(j)}$.

4. Resample the smoothed particles according to

$$\Pr\left(\left(x_{t+1|N}^{(i)}, x_{t|N}^{(i)}\right) = \left(x_{t+1|N}^{(j)}, x_{t|t}^{(j)}\right)\right) = \tilde{q}_{t|N}^{(j)}. \quad (25)$$

5. Set $t := t - 1$ and iterate from step 3.
-

5.4 Using a Particle Smoother with EM

In Lemmas F.1 and F.2 we require the computation of various expectations that are conditional on the data Y_N . In the following lemma we provide explicit formulations of these expectations in terms of smoothed particles as calculated in Algorithm F.4.

Lemma F.3

Using the smoothed state particles as calculated in Algorithm F.4 we have the following approximations

$$\mathbb{E}_{\theta_k} \{ \alpha_t^T \alpha_t | Y_N \} \approx \frac{1}{M} \sum_{i=1}^M \left(\alpha_t^{(i)} \right)^T \left(\alpha_t^{(i)} \right), \quad (26a)$$

$$\mathbb{E}_{\theta_k} \{ \alpha_t^T l_t | Y_N \} \approx \frac{1}{M} \sum_{i=1}^M \left(\alpha_t^{(i)} \right)^T \left(l_t^{(i)} \right), \quad (26b)$$

$$\mathbb{E}_{\theta_k} \{ x_1 | Y_N \} \approx \frac{1}{M} \sum_{i=1}^M x_{1|N}^{(i)}. \quad (26c)$$

Similarly,

$$\mathbb{E}_{\theta_k} \{ (x_1 - \mu)(x_1 - \mu)^T | Y_N \} \approx \frac{1}{M} \sum_{i=1}^M \left(x_{1|N}^{(i)} - \mu \right) \left(x_{1|N}^{(i)} - \mu \right)^T, \quad (26d)$$

$$\mathbb{E}_{\theta_k} \{ (l_t - \alpha_t \vartheta)(l_t - \alpha_t \vartheta)^T | Y_N \} \approx \frac{1}{M} \sum_{i=1}^M \left(l_t^{(i)} - \alpha_t^{(i)} \vartheta \right) \left(l_t^{(i)} - \alpha_t^{(i)} \vartheta \right)^T. \quad (26e)$$

where $l_t^{(i)}$ and $\alpha_t^{(i)}$ are simply the respective functions evaluated at the i^{th} particle $x_{t|N}^{(i)}$.

6 Simulation Example

This section profiles the performance of the EM-based estimation methods just presented by way of considering the following nonlinear system.

$$x_{t+1} = ax_t + b \frac{x_t}{1+x_t^2} + c \cos(1.2t) + w_t, \quad (27a)$$

$$y_t = dx_t^2 + e_t, \quad (27b)$$

where $a = 0.5$, $b = 25$, $c = 8$, $d = 0.05$, $w_t \sim \mathcal{N}(0, 10^{-2})$ and $e_t \sim \mathcal{N}(0, 10^{-2})$. In terms of the structure in (1) we make the following associations

$$\alpha_t = \begin{pmatrix} x_t & \frac{x_t}{1+x_t^2} & \cos(1.2t) & 0 \\ 0 & 0 & 0 & x_t^2 \end{pmatrix}, \quad (28a)$$

$$\beta_t = 0, \quad (28b)$$

$$\vartheta^T = (a \ b \ c \ d). \quad (28c)$$

This system has been extensively studied in the context of state estimation (Gordon et al., 1993, Kitagawa, 1996, 1998, Doucet et al., 2000, Godsill et al., 2004). However, it has not been the subject of great attention from the parameter estimation viewpoint of this paper.

As is well recognized (Ljung, 2003), a particularly important aspect of nonlinear system estimation is the difficulty of finding appropriate initial parameter values with which

to initialize an iterative search. To address this issue, and in so doing illustrate the inherent robustness of the EM-based approach presented here, each of the 200 simulation runs was initialized at a randomly chosen initial estimate $\hat{\theta}_0$ which itself was formed using perturbations from the true values.

Using $N = 1000$ data samples, and despite only using a very modest number of $M = 50$ particles in the smoothing calculations, the empirical estimation results shown in Figure 1 are encouraging. In particular, note that despite quite widely varying initial-

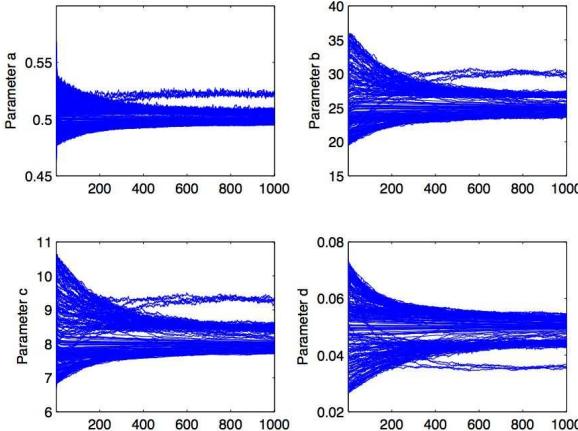


Figure 1: Parameter estimates for each of the 200 simulation runs as they evolve over 1000 iterations of the EM method. The true parameter values are $a = 0.5$, $b = 25$, $c = 8$ and $d = 0.05$.

izations, convergence to the true parameters occurred in most cases. Further simulations were conducted with $M = 100$ and higher number of particles, but without any significant performance benefit. This suggests a robustness of the EM-based approach to inaccuracies in computation in the E-step.

In relation to this, note that the method requires $\mathcal{O}(NM^2)$ floating point operations per iteration. The computational load is sensitive to the number of particles chosen, but scales well with increasing observed data length. To provide a reference point for these scaling comments, each simulation required to present the Monte Carlo presentation in Figure 1 completed within 3 minutes on a Pentium IV running at 3GHz.

By way of comparison, alternative methods, including Newton-based gradient search were also tried, but proved very unsuccessful. To explore the reason behind this, and also to emphasize the surprising robustness to initial starting point just presented, consider the simpler estimation problem which involves estimating only $\vartheta = (a, b)^T$ with c and d fixed to their true values, and with the additive noise w_t and e_t set to zero. The former is done so that the cost surface implied by the likelihood can be visualized, and the latter is considered so that attention is focused solely on how the nonlinear dynamics affects the difficulty of the estimation problem.

The resulting mean square error (the dominating component of the likelihood computation) cost surface is shown in Figure 2. Clearly, it is very far from convex. Note that the very irregular cost function, even if due to finite precision effects and not intrinsic, is

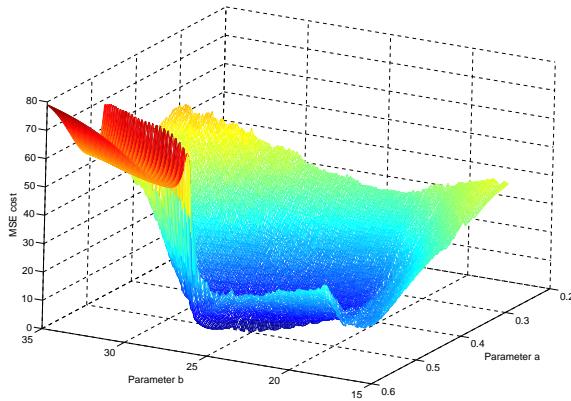


Figure 2: Surface plot of the MSE versus parameters a and b only.

still an obstacle to gradient based methods but not, as will be illustrated, to an EM-based approach. The perhaps surprising complexity from such a simple example underlines the particular difficulties of nonlinear system estimation.

The MSE cost function associated with the present problem contains quite a few local minima. It is therefore not surprising that gradient-based search was found to perform so poorly on the preceding example. To emphasize this, Figure 3 shows a contour plot of the the MSE cost function. Clearly, and as suggested in the previous figure, there seems to be a large number of local minima, any of which may attract gradient-based approaches. Indeed, the black lines shown in that diagram are Gauss – Newton gradient-based search trajectories for 25 different starting points, and all become locked in local minima. By way of contrast, Figure 4 shows the estimate trajectories of the EM-based algorithm of this paper. Note that from the same starting points, all cases converge to the global maximum.

7 Conclusion

This paper has explored an approach to nonlinear dynamic system estimation whose key distinguishing features include the use of EM-based methods as opposed to more traditional gradient-based search, a fairly general model structure, the use of Monte Carlo based particle methods for the computation of required smoothed state estimates, and a capacity for simply encompassing multivariable problems.

By way of example, the resulting approach has been demonstrated to be (perhaps) surprisingly robust to attraction to local minima, even in cases where the underlying cost is extremely “irregular” and non-convex. Further work is required to understand the mechanisms underlying this robustness, and to test the ideas on more substantial problem sizes.

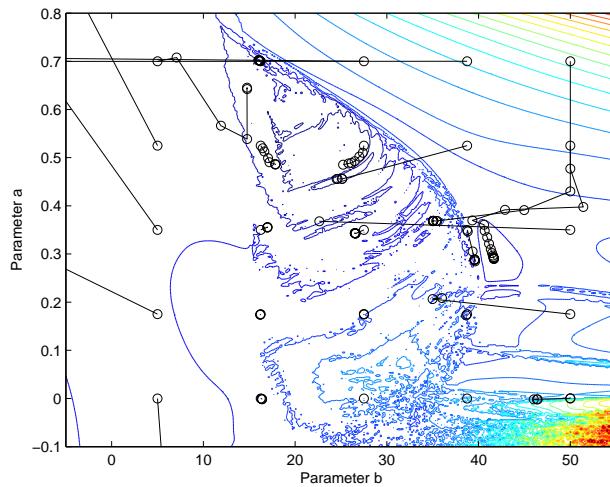


Figure 3: Contour plot of MSE cost for the case of identifying parameters a and b only, together with Gauss – Newton gradient-based search estimate trajectories overlaid. Note that, presumably due to the very large number of local minima, no trajectories converge to the global minimum.

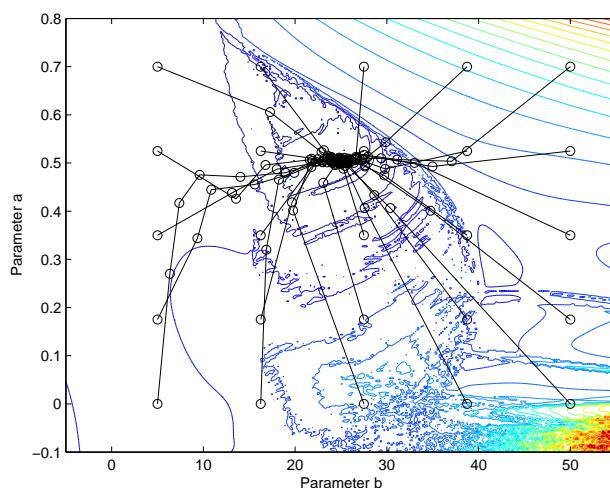


Figure 4: Same as previous plot, but with EM-based estimate trajectories for 25 different starting points. Note that all converge to the global minimum.

References

- Andrieu, C., Doucet, A., Singh, S. S., and Tadić, V. B. (2004). Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92(3):423–438.
- Bendat, J. S. (1990). *Nonlinear System Analysis and Identification from Random Data*. Wiley Interscience.
- Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.
- Doucet, A., de Freitas, N., and Gordon, N., editors (2001). *Sequential Monte Carlo Methods in Practice*. Springer Verlag.
- Doucet, A., Godsill, S. J., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.
- Doucet, A. and Tadić, V. B. (2003). Parameter estimation in general state-space models using particle methods. *Annals of the Institute of Statistical Mathematics*, 55:409–422.
- Gibson, S. and Ninness, B. (2005). Robust maximum-likelihood estimation of multivariable dynamic systems. *Automatica*, 41(10):1667–1682.
- Gibson, S., Wills, A., and Ninness, B. (2005). Maximum-likelihood parameter estimation of bilinear systems. *IEEE Transactions on Automatic Control*, 50(10):1581–1596.
- Godsill, S. J., Doucet, A., and West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113.
- Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25.
- Kitagawa, G. (1998). A self-organizing state-space model. *Journal of the American Statistical Association*, 93(443):1203–1215.
- Leontaritis, I. J. and Billings, S. A. (1985). Input-output parametric models for non-linear systems. part II: stochastic non-linear systems. *International Journal of Control*, 41(2):329–344.
- Liu, J. S. (1996). Metropolized independent sampling with comparison to rejection sampling and importance sampling. *Statistics and Computing*, 6:113–119.
- Ljung, L. (1999). *System identification, Theory for the user*. System sciences series. Prentice Hall, Upper Saddle River, NJ, USA, second edition.
- Ljung, L. (2003). Bode Lecture: Challenges of Nonlinear System Identification. *Proceedings of the 42nd IEEE Conference on Decision and Control*, Hawaii, USA.

- Ljung, L. and Vicino, A., editors (2005). Special Issue on System Identification, volume 50 (10). *IEEE Transactions on Automatic Control*.
- Narendra, K. and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1:4–27.
- Rangan, S., Wolodkin, G., and Poolla, K. (1995). New results for Hammerstein system identification. In *Proceedings of the 34th IEEE Conference on Decision and Control*, pages 697–702, New Orleans, USA.
- Schön, T. and Gustafsson, F. (2003). Particle filters for system identification of state-space models linear in either parameters or states. In *Proceedings of the 13th IFAC Symposium on System Identification*, pages 1287–1292, Rotterdam, The Netherlands. Invited paper.
- Söderström, T. and Stoica, P. (1989). *System identification*. Systems and Control Engineering. Prentice Hall.
- Tanizaki, H. (2001). Nonlinear and non-Gaussian state space modeling using sampling techniques. *Annals of the Institute of Statistical Mathematics*, 53(1):63–81.

Paper G

Integrated Navigation of Cameras for Augmented Reality

Authors: Thomas B. Schön and Fredrik Gustafsson

Edited version of paper originally published in *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, July 2005.

Preliminary version published as Technical Report LiTH-ISY-R-2645, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Integrated navigation of cameras for augmented reality

Thomas B. Schön and Fredrik Gustafsson

Department of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden.
E-mail: (schon, fredrik)@isy.liu.se.

Abstract

In augmented reality, the position and orientation of a camera must be estimated very accurately. This paper proposes a filtering approach, similar to integrated navigation in aircraft, which is based on inertial measurements as primary sensor on which dead-reckoning can be based, and features in the image as supporting information to stabilize the dead-reckoning. The image features are considered to be sensor signals in a Kalman filter framework.

Keywords: Sensor fusion, Kalman filter, inertial navigation, augmented reality, computer vision, feature extraction.

1 Introduction

THE idea in *augmented reality* (AR) is to add synthetic background and objects to streaming video images in real-time, while allowing the camera to move. One of the major technical challenges to achieve this is to determine the camera's position and orientation in 3D with very high accuracy and low latency. Typical applications of such a system includes studio recordings with synthetic scenes (Thomas et al., 1997) and virtual reconstruction of historical buildings (Vlahakis et al., 2002).

Prior work in this recent research area focuses on image processing algorithms, where the streaming image is the primary information source (Davison, 2003). This requires quite a lot of features in each image, and has lead to a development of markerbased systems, where bar-coded markers are installed in the studio (Thomas et al., 1997). Later work has tried to avoid artificial markers, by including other information like accelerations and angular velocities from inertial sensors (You et al., 1999, You and Neumann, 2001).

When it comes to using vision in AR two fundamentally different strategies have been used:

- The environment is prepared in advance using artificial markers, which impose a significant additional cost to these systems. Examples of this kind of system can be found in Caarls et al. (2003), Yokokohji et al. (2000), Thomas et al. (1997).
- Markerless systems, which use natural features occurring in the real scene as markers. The approach presented in this paper will utilize this strategy. The need for this kind of systems is motivated in Azuma et al. (1999). A common characteristic of these systems is that they use some kind of model of the scene. Some attempts to create such a system are given in You et al. (1999), Klein and Drummond (2003).

In this contribution, the reverse approach is applied. An inertial measurement unit (IMU) with three degrees of freedom accelerometers and gyroscopes is used as the primary source of information. Dead-reckoning gives a position and orientation relative to the initial camera location. This estimate will quite soon drift away and become completely useless, unless it is supported with secondary sensors, which in this case are provided by the images.

Our approach mimics the navigation systems in aircraft (Nordlund, 2002, Schön et al., 2005). There are obviously many similarities of aircraft navigation and our approach to augmented reality: the aircraft and camera have the same state vector, navigation is based on dead-reckoning IMU sensor signals, and both have to be supported by secondary information. For aircraft, infrastructure based positions from instrument landing systems or satellite positioning systems can be used. In military applications terrain navigation systems can be employed (Bergman et al., 1999). In this paper, features in the image are used as secondary sensors in two different ways:

- Feature displacement: An observed movement of a distinct feature in the image can be directly related to a movement in the camera, which will be shown to correspond to a one-dimensional measurement equation for each feature displacement.
- Recognition of known 3D-objects: Certain characteristic features in the scene are stored in a scene model prior to filtering. When such a feature is observed in the image, two degrees of freedom of the camera position can be determined.

A possible third direction is to use the homography (Hartley and Zisserman, 2003). However, this is not elaborated on within this paper. This idea has previously been discussed, see, e.g., Diel et al. (2005), Vidal et al. (2001).

By using the IMU as primary sensor, it is not necessary that all six degrees of freedom are present in the features in every image. This is the main advantage in the approach of fusing information from the inertial sensors with the information from the vision sensor.

2 Dynamic Motion Model

The dynamic state equations for the camera consist of a set of nonlinear differential equations describing how the camera pose is related to the readings from the accelerometers and the gyroscopes according to

$$\dot{x}(t) = f(x(t), u(t), t), \quad (1)$$

where the state vector $x(t)$ consists of position c_f , velocity v_f ($v_f = \dot{c}_f$) and orientation (represented using unit quaternions) q ($q = q_{cf}$), i.e., $x = (c_f^T, v_f^T, q^T)^T$, where

$$c_f = \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix}, \quad v_f = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}, \quad q = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}. \quad (2)$$

Regarding the notation, c_f is used to describe the position of the camera center (point C in Figure 1) expressed in the F -system. Furthermore, the accelerometer, a_c , and the gyroscope, ω_c , readings are considered to be input variables, u , i.e.,

$$u = \begin{pmatrix} a_c \\ \omega_c \end{pmatrix}. \quad (3)$$

In the subsequent sections the relevant coordinate systems are defined and the nonlinear function $f(\cdot)$ in (1) will be derived.

2.1 Geometry and Co-ordinate Systems

The following three coordinate systems are used:

1. **Fixed (F):** This is considered to be an inertial system (the rotation of the earth is ignored), which is fixed to earth. The navigation will be performed in this system. Furthermore, the scene model is given in this system.
2. **Camera (C):** This coordinate system is attached to the camera and hence it is moving with the camera. Its origin is located in the camera center.
3. **Image (I):** The image is projected into this coordinate system, which is located in the camera center.

These three coordinate systems are illustrated in Figure 1. Furthermore, a fourth coordinate system, the sensor system, is used. This is the coordinate system in which the inertial measurements are obtained. It is not discussed in this paper, which implies that a somewhat unrealistic assumption is used, namely that the inertial sensors are placed in the camera center. However, everything discussed in this paper can rather straightforwardly be adapted to the fact that the sensor coordinate system is present as well.

2.2 Position

The position of the camera is given by the position of the camera center (point C in Figure 1). The accelerometers measures the inertial forces w.r.t. an inertial system (the F -system in this work). Hence, the accelerometers will measure the difference between the acceleration of the camera, a_f , and the gravity vector, g_f . However, since the accelerometers are attached to the camera (strapdown inertial system) the measurements will be resolved in the camera coordinate system, according to

$$a_c = R_{cf} (a_f - g_f), \quad (4)$$

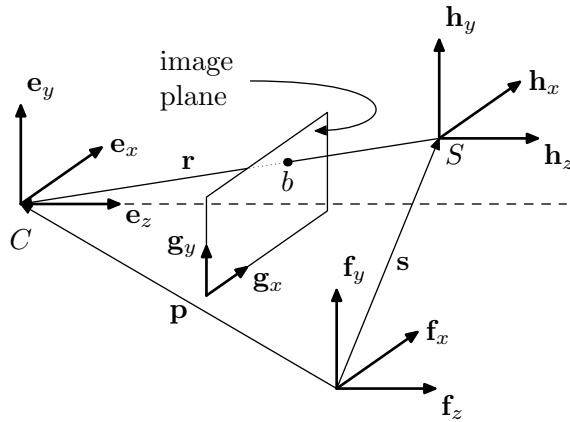


Figure 1: Illustration of the different coordinate systems and how they are related. Point C is the position of the camera (optical center) and point S is the position of a certain static feature in the real scene.

where R_{cf} is a rotation matrix which rotates vectors from the F -system to the C -system. Notice that the accelerometer measurement can be modeled as a measurement signal (Rehbinder and Hu, 2004), $y = a_c$, or as an input signal, $u = a_c$, (common in the aircraft industry). In this work the accelerometer signal is modeled as an input signal, in order to avoid additional states. However, by including the acceleration and the angular velocity in the state vector the acceleration and the angular velocity can be modeled by shaping the process noises for these states. The dynamic motion model is according to Newton's second law a double integration of the measured acceleration:

$$\dot{c}_f = v_f, \quad (5a)$$

$$\ddot{v}_f = R_{fc}a_c + g_f. \quad (5b)$$

By assuming that the input signal is piecewise constant it is straightforward to derive a discrete-time version of (5).

2.3 Orientation

Finding a suitable representation for the orientation of a rigid body in 3D is a more intricate problem than one might first guess. In Section 2.2 rotation matrices (commonly referred to as Direction Cosine Matrices (DCM)) were used to describe rotations. These matrices belong to a group called $SO(3)$, defined by

$$SO(3) = \{R \in \mathbf{R}^{3 \times 3} : RR^T = I, \det R = +1\}. \quad (6)$$

The name SO stands for *special, orthogonal*, due to the constraints (6) (Murray et al., 1994). Hence, the most natural description to use is DCM. However, this description has some problems, since it requires six parameters and since it is hard to enforce the orthogonality condition. It has been shown that five is the minimum number of parameters that

have to be used in order to parameterize the rotation group in such a way that a global description, without singular points is obtained (Hopf, 1940, Stuelpnagel, 1964). However, the dynamics for this parameterization is quite complicated, which implies that it is not used. Using four parameters, unit quaternions¹, to describe the orientation provides the best alternative, since it is a representation that is nonsingular and the dynamics is linear (bilinear if the angular velocity is modeled as a state variable) in the states. The downside is that the unit constraint has to be maintained and that the parameterization is non-global. However, this non-global property will not be a problem in practice. Another commonly used parameterization is the Euler angles. The advantage of this parameterization is that it only requires three parameters, but the dynamics is nonlinear and it is a singular, non-global representation. According to the authors the best trade-off for parameterizing the rotation group is provided by the unit quaternion. Hence, all computations are performed using unit quaternions. However, when the orientation is presented to the user Euler angles are used, since this parameterization is the easiest to interpret and visualize.

A good account of the twelve most common rotation parameterizations is given in Shuster (1993). Furthermore, Shoemake (1985) provides some good intuition regarding the unit quaternions. The dynamic equation for the quaternions is

$$\dot{q}(t) = \frac{1}{2}\Omega(\omega)q(t), \quad (7)$$

where

$$\Omega(\omega) = \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix}. \quad (8)$$

The quaternion has to be normalized, i.e.,

$$q^T(t)q(t) = 1, \quad (9)$$

in order to represent an orientation. By invoking the assumption that the angular velocity is constant between the sampling instants the rotation vector θ can be defined as

$$\theta = \omega_t T_s, \quad (10)$$

and under this assumption it can be shown that the solution to (7) is

$$q_{t+1} = A(\theta)q_t, \quad (11)$$

where $A(\theta)$ can be shown to be

$$A(\theta) = \cos(\|\theta\|/2) I_4 + \frac{\sin(\|\theta\|/2)}{\|\theta\|} \Omega(\theta). \quad (12)$$

Care has to be taken when estimating the orientation, since the set of all rotations, SO(3) is not a vector space, but rather a manifold, due to the constraint (6). Using quaternions this is handled simply by normalizing the estimate. However, the best would of course be if an estimator could be derived that delivered estimates, which inherently existed on the manifold. The problem is that the resulting problem is non-convex.

¹Another name for the unit quaternion is Euler-Rodrigues symmetric parameters, or Euler symmetric parameters (Shuster, 1993).

3 Using Vision as a Sensor

In order to be able to incorporate the information available in the image sequence into the estimation problem measurement equations

$$y_t = h(x_t, e_t, t), \quad (13)$$

have to be derived. These equations should describe the relationship between the state variable x and the information available in the images. In the subsequent sections two different approaches on how to derive these equations are discussed. Since a single image contains large amounts of information the most essential information has to be efficiently recovered. The approach used in this work is to extract features from the images. In the computer vision literature an *image feature* is any structural feature that can be extracted from the image. The idea of using inertial sensors and features extracted from the images have previously been exploited, e.g., in Rehbinder and Ghosh (2003), Jiang et al. (2004).

3.1 Camera Model

A camera is a device that provides two dimensional projections of a three dimensional real scene. The camera model describes this projection in mathematical terms. Hence, the camera model is most essential in forming the measurement equations. The camera model used in this work is the *pinhole model* (Hartley and Zisserman, 2003),

$$(x_i, y_i)^T = (fx/z, fy/z)^T, \quad (14)$$

where $(x_i, y_i)^T$ are the coordinates for the feature in the image coordinate system, see Figure 1. Furthermore, $(x, y, z)^T$ is the corresponding position in the real scene and f is the focal length. The model (14) is simply a way to state that two objects lying on the same ray will be projected onto the same point in the image plane. This model is used due to its simplicity. However, all equations derived in this paper can be extended to more advanced camera models including parameters for optical distortion etc. For more details on different camera models, see Hartley and Zisserman (2003).

3.2 Two Dimensional Feature Displacements

The goal of this section is to derive a measurement equation using the fact that an observed movement of a static feature in the image can be directly related to a movement of the camera. This measurement equation can then be used within the Kalman filtering framework. The derivation starts with the simple fact that,

$$\mathbf{r} = \mathbf{r} \Leftrightarrow 0 = \mathbf{r} + x\mathbf{e}_x + y\mathbf{e}_y + z\mathbf{e}_z, \quad (15)$$

where the \mathbf{r} -vector, defined in Figure 1, has been expressed in two different coordinate systems, the rotating C -system and the fixed feature system. Differentiating (15) w.r.t. time gives

$$0 = \dot{\mathbf{r}} + \dot{x}\mathbf{e}_x + \dot{y}\mathbf{e}_y + \dot{z}\mathbf{e}_z + x\dot{\mathbf{e}}_x + y\dot{\mathbf{e}}_y + z\dot{\mathbf{e}}_z. \quad (16)$$

From Figure 1; $\dot{\mathbf{r}} = \dot{\mathbf{p}} - \dot{\mathbf{s}} = \dot{\mathbf{p}}$, which together with the fact that $x\dot{\mathbf{e}}_x + y\dot{\mathbf{e}}_y + z\dot{\mathbf{e}}_z = \boldsymbol{\omega} \times \mathbf{r}$ gives

$$\dot{x} = -z\omega_y + y\omega_z - v_x, \quad (17a)$$

$$\dot{y} = -x\omega_z + z\omega_x - v_y, \quad (17b)$$

$$\dot{z} = -y\omega_x + x\omega_y - v_z, \quad (17c)$$

where $\mathbf{v} = \dot{\mathbf{p}}$. Differentiating (14) gives

$$\dot{x}_i = f \frac{\dot{x}z - x\dot{z}}{z^2}, \quad (18a)$$

$$\dot{y}_i = f \frac{\dot{y}z - y\dot{z}}{z^2}. \quad (18b)$$

Inserting (14) and (17) in (18) gives

$$\dot{x}_i = \underbrace{\frac{x_i y_i}{f} \omega_x - f \left(1 + \frac{x_i^2}{f^2} \right) \omega_y + y_i \omega_z}_{\dot{x}_{i,R}} + \underbrace{\frac{-fv_x + x_i v_z}{z}}_{\dot{x}_{i,T}}, \quad (19a)$$

$$\dot{y}_i = \underbrace{f \left(1 + \frac{y_i^2}{f^2} \right) \omega_x - \frac{x_i y_i}{f} \omega_y - x_i \omega_z}_{\dot{y}_{i,R}} + \underbrace{\frac{-fv_y + y_i v_z}{z}}_{\dot{y}_{i,T}}, \quad (19b)$$

where the velocity has been split into one rotational part (indicated with subscript R), and one translational part (indicated with subscript T). It is impossible to use (19) to gain perfect information about the present position and orientation of the camera, which has previously been discussed in You et al. (1999), Matthies et al. (1988), Longuet-Higgins and Prazdny (1980). However, in combination with the other sensors these equations will help in the task of finding the position and orientation of the camera. Gyroscopes provide measurements of the angular velocity $\boldsymbol{\omega}$ and hence the rotational terms in (19) can be considered to be known (with a certain degree of uncertainty).

The measurements are the projection of the features in the image plane, i.e.,

$$y^j = \begin{pmatrix} x_i^j \\ y_i^j \end{pmatrix} + e_t^j, \quad j = 1, \dots, N, \quad (20)$$

where N is the number of features, and e the measurement noise. However, since (19) is used the measurement equations will be implicit, i.e., the measurement equations will not be in the form (13), but rather in the following form:

$$0 = h(y_t, y_{t-1}, x_t, e_t, t). \quad (21)$$

There is one problem with the derived measurement equations, the depth information z of the feature is still present. This problem can be tackled in numerous ways. The first idea that comes to mind is to extend the state vector with the depth states $z^i, i = 1, \dots, N$. In Davison (2003) an algorithm similar to the particle filter is used to estimate

the depths. Alternatively, the depths can be thought of as *nuisance* variables which should be eliminated. Substituting z from (19a) into (19b) gives

$$\frac{\dot{y}_i - \dot{y}_{i,R}}{\dot{x}_i - \dot{x}_{i,R}} = \frac{-fv_y + y_i v_z}{-fv_x + x_i v_z}, \quad (22)$$

which is the resulting one-dimensional measurement equation. It is straightforward to rewrite (22) on the form (21), using the Euler approximation for the differential operator, according to

$$\alpha_t (-fv_{x,t} + x_{i,t}v_{z,t}) = -fv_{y,t} + x_{i,t}v_{z,t}, \quad (23)$$

where

$$\alpha_t = \left(\frac{y_{i,t} - y_{i,t-1}}{T_s} - \dot{y}_{i,R} \right) / \left(\frac{x_{i,t} - x_{i,t-1}}{T_s} - \dot{x}_{i,R} \right).$$

Finally (23) can be written

$$\underbrace{(-f\alpha_t \quad f \quad (\alpha_t x_{i,t} - y_{i,t}))}_{C_t} v_f = 0. \quad (24)$$

This is the resulting measurement equation for two dimensional feature displacement.

3.3 Three Dimensional Features and Model

The vision system delivers a list of N feature coordinates in the image plane, $\{x_i^j, y_i^j\}_{j=1}^N$ and the corresponding positions, $\{s_j = (s_{j,x}, s_{j,y}, s_{j,z})\}_{j=1}^N$, in the real scene. This position is obtained from a three dimensional model of the world in which the camera is moving. This model is generated off-line. Intuitively, this information should provide valuable information for estimating the camera pose. Using (14) and Figure 1 gives

$$(x_i, y_i)^T = (fr_{c,x}/r_{c,z}, fr_{c,y}/r_{c,z})^T, \quad (25)$$

where r_c is the vector from the camera center to the current feature. Figure 1 also reveals that

$$r_c = R_{cf} (c_f - s_f). \quad (26)$$

The resulting measurement equation is found by using the idea from the previous section, i.e., writing the measurement equation in the implicit form (21). This results in

$$0 = \begin{pmatrix} r_{c,z}x_i - fr_{c,x} \\ r_{c,z}y_i - fr_{c,y} \end{pmatrix} + e, \quad (27)$$

which simply corresponds to multiplying (25) with $r_{c,z}$. Similar ideas have been presented in, e.g., Davison (2003). The difference is that in this work an off-line model of the real scene is used in combination with information from inertial sensors. Hence, the costly procedure of preparing the environment with artificial markers is not necessary. Furthermore, the information from the inertial sensors is obtained at a higher frequency than the vision measurements and will be especially helpful during fast maneuvers.

4 Illustration

In order to evaluate the ideas proposed in this paper a virtual environment, briefly described in this section, is used. More specifically a three dimensional model of a car is used. Figure 2 provides two images from the video sequence. The car is standing still and the camera is moving around the car in such a way that the camera is always facing the car. Since the true position and orientation of the camera is known, the acceleration

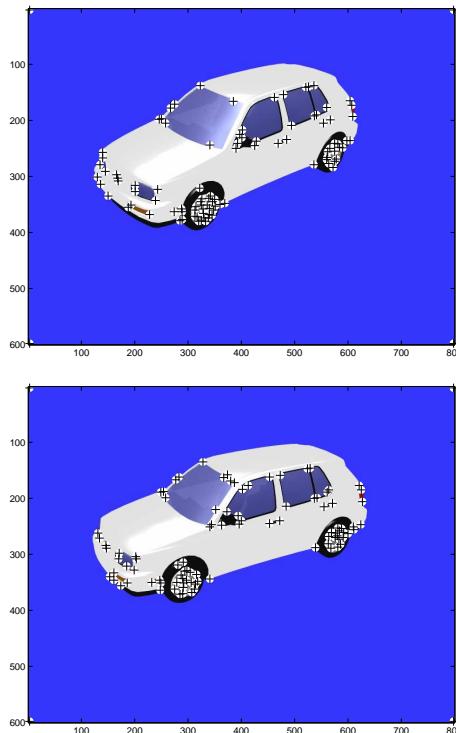


Figure 2: Two images from the video stream used to obtain the vision measurements. Furthermore, several 2D features have been indicated in the images. The camera has been rotated 10° from the upper to the lower image.

and angular velocity can be calculated. Using this the inertial measurements can be simulated, simply by adding the proper noise to the true accelerations and angular velocities. Furthermore, the 3D model of the car provides an image sequence from which features can be extracted. These features will constitute the vision measurements, which will be included in the estimation problem using the ideas discussed in Section 3.

The next step is to use authentic inertial and vision measurements, which will be provided by our partners at BBC R&D in London. They have a positioning system installed in their studio (referred to as free-D (Thomas et al., 1997)), providing the true pose, which can be used to assess the estimation performance. The authors are currently working together with Xsens (2005) on using the idea presented in Section 3.3. The preliminary results looks promising.

5 Conclusion

This paper propose a filtering approach for estimating the position and orientation of a camera in three dimensions. The underlying idea of supporting inertial sensors using additional sensors has previously been successfully used for instance within the aircraft industry. The difference is that in this work vision is used, instead of for instance terrain elevation databases, to support the dead-reckoning of the inertial sensor information. Furthermore, two different strategies regarding the process of incorporating vision measurements in the Kalman filtering framework were discussed. Finally, some brief illustrations on how to evaluate these ideas were given.

6 Acknowledgment

This work has been performed within the MATRIS consortium, which is a sixth framework research program within the European Union (EU), contract number: IST-002013. The authors would like to thank the EU for the financial support and the partners within the consortium for a fruitful collaboration this far. For more information about the MATRIS consortium, please visit the consortium home-page, www.ist-matrism.org.

References

- Azuma, R., Lee, J. W., Jiang, B., Park, J., You, S., and Neumann, U. (1999). Tracking in unprepared environments for augmented reality systems. *Computers & Graphics*, 23(6):787–793.
- Bergman, N., Ljung, L., and Gustafsson, F. (1999). Terrain navigation using Bayesian statistics. *IEEE Control Systems Magazine*, 19(3):33–40.
- Caarls, J., Jonker, P., and Persa, S. (2003). *Lecture Notes in Computer Science*, volume 2875, chapter Sensor Fusion for Augmented Reality, pages 160–176. Springer Verlag, Berlin, Germany.
- Davison, A. J. (2003). Real-time simultaneous localisation and mapping with a single camera. In *Proceedings Ninth IEEE International Conference on Computer Vision*, volume 2, pages 1403–1410, Nice, France.
- Diel, D. D., DeBitetto, P., and Teller, S. (2005). Epipolar constraints for vision-aided inertial navigation. In *Proceedings of the 7th IEEE Workshop on Motion and Video Computing*, volume 2, pages 221–228, Colorado, USA.
- Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in computer vision*. Cambridge University Press, Cambridge, UK, second edition.
- Hopf, H. (1940). Systeme symmetrischer bilinearformen und euklidische modelle der projektiven räume. *Vierteljahrsschrift der Naturf. Gesellschaft in Zürich*, 85:165–177.
- Jiang, B., Neumann, U., and You, S. (2004). A robust hybrid tracking system for outdoor augmented reality. In *Virtual Reality 2004*, pages 3–10, Chicago, USA.

- Klein, G. and Drummond, T. (2003). Robust visual tracking for non-instrumental augmented reality. In *International Symposium on Mixed Augmented Reality*, pages 113–123, Tokyo, Japan.
- Longuet-Higgins, H. C. and Prazdny, K. (1980). The interpretation of a moving retinal image. *Proceedings of the Royal Society of London, Series B, Biological Sciences*, 208(1173):385–397.
- Matthies, L., Szeliski, R., and Kanade, T. (1988). Kalman filter-based algorithms for estimating depth from image sequences. Technical Report CMU-RI-TR-88-1, The Robotics Institute, Carnegie Mellon University.
- Murray, R. M., Li, Z., and Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.
- Nordlund, P.-J. (2002). *Sequential Monte Carlo Filters and Integrated Navigation*. Licentiate Thesis No 945, Department of Electrical Engineering, Linköping University, Sweden.
- Rehbinder, H. and Ghosh, B. K. (2003). Pose estimation using line-based dynamic vision and inertial sensors. *IEEE Transactions on Automatic Control*, 48(2):186–199.
- Rehbinder, H. and Hu, X. (2004). Drift-free attitude estimation for accelerated rigid bodies. *Automatica*, 40(4):653–659.
- Schön, T., Gustafsson, F., and Nordlund, P.-J. (2005). Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289.
- Shoemake, K. (1985). Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, San Francisco, USA.
- Shuster, M. D. (1993). A survey of attitude representations. *The Journal of the Australasian Sciences*, 41(4):439–517.
- Stuepnagel, J. (1964). On the parameterisation of the three-dimensional rotation group. *SIAM Review*, 6(4):422–430.
- Thomas, G. A., Jin, J., Niblett, T., and Urquhart, C. (1997). A versatile camera position measurement system for virtual reality tv production. In *International Broadcasting Convention*, pages 284–289, Amsterdam, The Netherlands.
- Vidal, R., Ma, Y., Hsu, S., and Sastry, S. (2001). Optimal motion estimation from multiview normalized epipolar constraint. In *8th International Conference on Computer Vision*, volume 1, pages 34–41, Vancouver, Canada.
- Vlahakis, V., Ioannidis, N., Karigiannis, J., Tsotros, M., Gounaris, M., Stricker, D., Gleue, T., Daehne, P., and Almeida, L. (2002). Archeoguide: An augmented reality guide for archaeological sites. *Computer Graphics in Art History and Archaeology*, 22(5):52–60.

Xsens (2005). Xsens Technologies, Enschede, The Netherlands. www.xsens.com.

Yokokohji, Y., Sugawara, Y., and Yoshikawa, T. (2000). Accurate image overlay on video see-through HMDs using vision and accelerometers. In *Virtual Reality 2000 Conference*, pages 247–254, New Brunswick, NJ USA.

You, S. and Neumann, U. (2001). Fusion of vision and gyro tracking for robust augmented reality registration. In *IEEE Virtual Reality*, pages 71–78, Yokohama, Japan.

You, S., Neumann, U., and Azuma, R. (1999). Orientation tracking for outdoor augmented reality registration. *IEEE Computer Graphics and Applications*, 19(6):36–42.

Paper H

The Marginalized Particle Filter in Practice

Authors: Thomas B. Schön, Rickard Karlsson, and Fredrik Gustafsson.

Edited version of paper accepted for publication in *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, USA, March 2006.

Preliminary version published as Technical Report LiTH-ISY-R-2715, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

The Marginalized Particle Filter in Practice

Thomas B. Schön, Rickard Karlsson, and Fredrik Gustafsson

Department of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden.
E-mail: (schon, rickard, fredrik)@isy.liu.se.

Abstract

The marginalized particle filter is a powerful combination of the particle filter and the Kalman filter, which can be used when the underlying model contains a linear sub-structure, subject to Gaussian noise. This paper will illustrate several positioning and target tracking applications, solved using the marginalized particle filter. Furthermore, we analyze several properties of practical importance, such as its computational complexity and how to cope with quantization effects.

Keywords: Nonlinear state estimation, marginalized particle filter, positioning applications, target tracking applications.

1 Introduction

MANY problems in for instance positioning and target tracking can be cast as nonlinear state estimation problems, where the uncertainty in the process model and/or in the measurement model may be non-Gaussian. Such a general model can be formulated according to

$$x_{t+1} = f(x_t, u_t) + w_t, \quad (1a)$$

$$y_t = h(x_t) + e_t, \quad (1b)$$

with state variable $x_t \in \mathbf{R}^m$, input signal u_t and measurements $Y_t = \{y_i\}_{i=1}^t$, with known probability density functions for the process noise $p_w(w)$ and the measurement noise $p_e(e)$. Hence, traditional estimation methods based on the *Kalman filter* (KF) (Kalman, 1960, Kailath et al., 2000), or linearized version thereof, do not always provide good performance. Over the past 40 years there has been several suggestions on how to tackle the problem of estimating the states in (1). An appealing solution is provided by the *particle filter* (PF) (Gordon et al., 1993, Doucet et al., 2001a, Ristic et al., 2004),

which allows for a systematic treatment of both nonlinearities and non-Gaussian noise. However, due to the inherent computational complexity of the particle filter, real-time issues arise in many applications when the sampling rate is high. If the model includes a sub-structure with linear equations, subject to Gaussian noise, it is often possible to perform the estimation more efficiently. Here, this method is referred to as the *marginalized particle filter* (MPF), it is also known as the Rao-Blackwellized particle filter, see ,for instance, Doucet et al. (2000, 2001a), Chen and Liu (2000), Andrieu and Doucet (2002), Andrieu and Godsill (2000), Schön et al. (2005). The MPF is a clever combination of the standard particle filter and the Kalman filter. It is a well known fact that in some cases it is possible to obtain better estimates, i.e., estimates with reduced variance, using the marginalized particle filter instead of using the standard particle filter (Doucet et al., 2001b).

The aim of this paper is to explain how the marginalized particle filter works in practice. We will try to achieve this by considering several applications where we have successfully applied the MPF. Since we cannot cover all the details in this paper references to more detailed descriptions are provided. Furthermore, the algorithm's computational complexity and the presence of quantization effects are analyzed, due to their importance in practical applications. To summarize, the analysis and applications covered are

Theory and analysis:

- *Background theory*
- *Complexity analysis*
- *Quantization effects*

Positioning applications:

- *Underwater terrain-aided positioning*
- *Aircraft terrain-aided positioning*
- *Automotive map-aided positioning*

Target tracking applications:

- *Automotive target tracking*
- *Bearings-only target tracking*
- *Radar target tracking*

There are certainly more applications of the marginalized particle filter reported in the literature. Just to mention a few, there are communication applications (Chen et al., 2000, Wang et al., 2002), nonlinear system identification (Schön and Gustafsson, 2003, Li et al., 2003, Daly et al., 2005), GPS navigation (Giremus and Tourneret, 2005) and audio source separation (Andrieu and Godsill, 2000).

The paper is organized as follows. In Section 2, the background theory and MPF algorithm are briefly introduced. The algorithm performance, computational complexity and ability to handle quantization effects are analyzed in Section 3. In Section 4, the applications are introduced and the structure of the underlying models is reviewed. The positioning and target tracking application are described in more detail in Section 5 and Section 6, respectively. Finally, Section 7 provides a concluding discussion of some lessons learned in using the marginalized particle filter.

2 Marginalized Particle Filter

The aim of recursively estimating the filtering density $p(x_t|Y_t)$ can be accomplished using the standard particle filter. However, if there is a linear sub-structure, subject to Gaussian noise, present in the model this can be exploited to obtain better estimates and possibly reduce the computational demand as well. This is the motivation underlying the marginalized particle filter.

2.1 Representation

The task of nonlinear filtering can be split into two parts: representation of the filtering probability density function and propagation of this density during the time and measurement update stages. Figure 1 illustrate different representations of the filtering density for a two-dimensional example. The *extended Kalman filter* (EKF) (Anderson and Moore, 1979, Kailath et al., 2000), can be interpreted as using one Gaussian distribution for representation and the propagation is performed according to a linearized model. The *Gaussian sum filter* (Anderson and Moore, 1979, Sorenson and Alspach, 1971) extends the EKF to be able to represent multi-modal distributions, still with an approximate propagation.

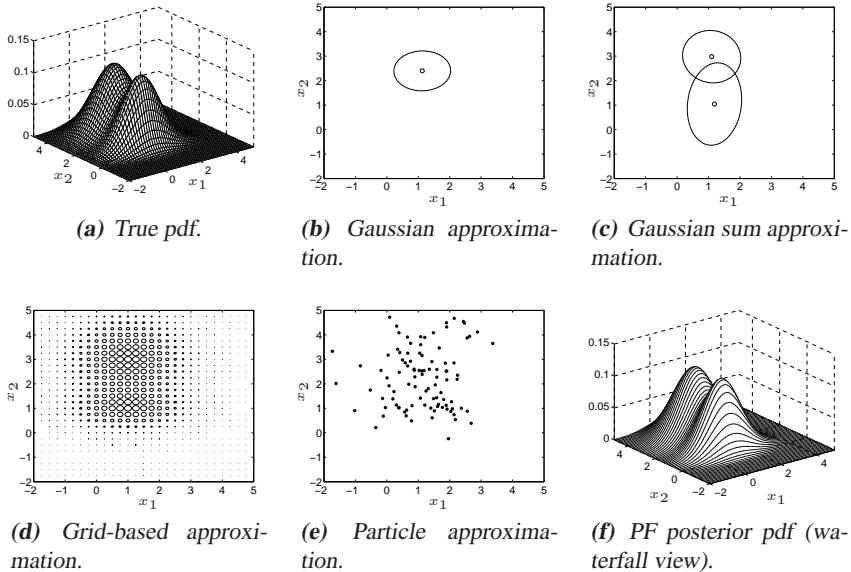


Figure 1: True probability density function and different approximate representations, in order of appearance, Gaussian, Gaussian sum, point-masses (grid-based approximation), particle samples and waterfall view that corresponds to the MPF.

Figure 1(d)–(f) illustrates numerical approaches where the exact nonlinear relations present in the model are used for propagation. The *point-mass filter* (grid-based approximation) (Bergman, 1999) employ a regular grid, where the grid weight is proportional to

the posterior. The *particle filter* (PF), (Gordon et al., 1993) represents the posterior by a stochastic grid in form of a set of samples, where all particles (samples) have the same weight. Finally, the *marginalized particle filter* (MPF) uses a stochastic grid for some of the states, and Gaussian distributions for the rest. That is, the MPF can be interpreted as a particle representation for a subspace of the state dimension, where each particle has an associated Gaussian distribution for the remaining state dimensions. This is the *waterfall* view in Figure 1(f). It will be demonstrated that an exact nonlinear propagation is still possible if there is a linear sub-structure in the model. An important model class has the property that the (co-)variance is the same for all particles, which simplifies computations significantly.

2.2 Model

Consider a state vector x_t , which can be partitioned according to

$$x_t = \begin{pmatrix} x_t^l \\ x_t^n \end{pmatrix}, \quad (2)$$

where x_t^l denotes the linear states and x_t^n denotes the nonlinear states, in the dynamics and measurement relation. A rather general model with the properties discussed above is given by

$$x_{t+1}^n = f_t^n(x_t^n) + A_t^n(x_t^n)x_t^l + G_t^n(x_t^n)w_t^n, \quad (3a)$$

$$x_{t+1}^l = f_t^l(x_t^n) + A_t^l(x_t^n)x_t^l + G_t^l(x_t^n)w_t^l, \quad (3b)$$

$$y_t = h_t(x_t^n) + C_t(x_t^n)x_t^l + e_t, \quad (3c)$$

where the state noise is assumed white and Gaussian distributed with

$$w_t = \begin{pmatrix} w_t^l \\ w_t^n \end{pmatrix} \sim \mathcal{N}(0, Q_t), \quad Q_t = \begin{pmatrix} Q_t^l & Q_t^{ln} \\ (Q_t^{ln})^T & Q_t^n \end{pmatrix}. \quad (3d)$$

The measurement noise is assumed white and Gaussian distributed according to

$$e_t \sim \mathcal{N}(0, R_t). \quad (3e)$$

Furthermore, x_0^l is Gaussian,

$$x_0^l \sim \mathcal{N}(\bar{x}_0, \bar{P}_0). \quad (3f)$$

Finally, the density of x_0^n can be arbitrary, but it is assumed known. More specifically, conditioned on the nonlinear state variables there is a linear sub-structure, subject to Gaussian noise available in (3), given by (3b).

2.3 Algorithm

Bayesian estimation methods, such as the particle filter, provide estimates of the filtering density function $p(x_t|Y_t)$. By employing the fact

$$p(x_t^l, X_t^n|Y_t) = p(x_t^l|X_t^n, Y_t)p(X_t^n|Y_t), \quad (4)$$

we can put the problem in a description suitable for the MPF framework, i.e., to analytically marginalize out the linear state variables from $p(x_t|Y_t)$. Note that $p(x_t^l|X_t^n, Y_t)$ is analytically tractable, since X_t^n is given. Hence, the underlying model is linear, Gaussian, and the pdf can be computed from the Kalman filter. Furthermore, an estimate of $p(X_t^n|Y_t)$ is provided by the particle filter. These two algorithms can then be combined into a single algorithm, the marginalized particle filter. Another name for this technique is the Rao-Blackwellized particle filter, and it has been known for quite some time, see, e.g., Doucet et al. (2000), Casella and Robert (1996), Doucet et al. (2001b), Chen and Liu (2000), Andrieu and Doucet (2002), Doucet et al. (2001b), Schön et al. (2005), Nordlund (2002). If the same numbers of particles are used in the standard particle filer and the marginalized particle filter, the latter will provide estimates of better or at least the same quality. Intuitively this makes sense, since the dimension of $p(x_t^n|Y_t)$ is smaller than the dimension of $p(x_t|Y_t)$, implying that the particles occupy a lower dimensional space. Furthermore, the optimal algorithm is used to estimate the linear state variables. For a detailed discussion regarding the improved accuracy of the estimates, see, e.g., Doucet et al. (1999, 2001b).

The marginalized particle filter for estimating the states in a dynamic model in the form (3) is provided in Algorithm H.1.

Algorithm H.1 (Marginalized particle filter)

1. *Initialization:* For $i = 1, \dots, N$, initialize the particles, $x_{0|-1}^{n,(i)} \sim p_{x_0^n}(x_0^n)$ and set $\{x_{0|-1}^{l,(i)}, P_{0|-1}^{(i)}\} = \{\bar{x}_0^l, \bar{P}_0\}$. Set $t := 0$.
2. *Particle filter measurement update:* For $i = 1, \dots, N$, evaluate the importance weights

$$q_t^{(i)} = p\left(y_t|X_t^{n,(i)}, Y_{t-1}\right), \quad (5)$$

and normalize $\tilde{q}_t^{(i)} = q_t^{(i)} / \sum_{j=1}^N q_t^{(j)}$.

3. Resample N particles, with replacement,

$$\Pr\left(x_{t|t}^{n,(i)} = x_{t|t-1}^{n,(j)}\right) = \tilde{q}_t^{(j)}.$$

4. *Particle filter time update and Kalman filter:*

- (a) *Kalman filter measurement update:*

$$\hat{x}_{t|t}^l = \hat{x}_{t|t-1}^l + K_t \left(y_t - h_t - C_t \hat{x}_{t|t-1}^l \right), \quad (6a)$$

$$P_{t|t} = P_{t|t-1} - K_t M_t K_t^T, \quad (6b)$$

$$M_t = C_t P_{t|t-1} C_t^T + R_t, \quad (6c)$$

$$K_t = P_{t|t-1} C_t^T M_t^{-1}. \quad (6d)$$

- (b) *Particle filter time update (prediction):* For $i = 1, \dots, N$, predict new particles,

$$x_{t+1|t}^{n,(i)} \sim p\left(x_{t+1|t}^n | X_t^{n,(i)}, Y_t\right).$$

(c) Kalman filter time update:

$$\hat{x}_{t+1|t}^l = \bar{A}_t^l \hat{x}_{t|t}^l + G_t^l (Q_t^{ln})^T (G_t^n Q_t^n)^{-1} z_t + f_t^l + L_t (z_t - A_t^n \hat{x}_{t|t}^l), \quad (7a)$$

$$P_{t+1|t} = \bar{A}_t^l P_{t|t} (\bar{A}_t^l)^T + G_t^l \bar{Q}_t^l (G_t^l)^T - L_t N_t L_t^T, \quad (7b)$$

$$N_t = A_t^n P_{t|t} (A_t^n)^T + G_t^n Q_t^n (G_t^n)^T, \quad (7c)$$

$$L_t = \bar{A}_t^l P_{t|t} (A_t^n)^T N_t^{-1}, \quad (7d)$$

where

$$z_t = x_{t+1}^n - f_t^n, \quad (8a)$$

$$\bar{A}_t^l = A_t^l - G_t^l (Q_t^{ln})^T (G_t^n Q_t^n)^{-1} A_t^n, \quad (8b)$$

$$\bar{Q}_t^l = Q_t^l - (Q_t^{ln})^T (Q_t^n)^{-1} Q_t^{ln}. \quad (8c)$$

5. Set $t := t + 1$ and iterate from step 2.

Since the focus of the present paper is on the practical aspects of Algorithm H.1, we will merely provide the intuition for this algorithm here. For a detailed derivation, see Schön et al. (2005). From this algorithm, it should be clear that the only difference from the standard particle filter is that the time update (prediction) stage has been changed. In the standard particle filter, the prediction stage is given solely by step 4(b) in Algorithm H.1.

Let us now briefly discuss step 4 in Algorithm H.1. Step 4(a) is a standard Kalman filter measurement, update using the information available in the measurement y_t . Once this has been performed the new estimates of the linear states can be used to obtain a prediction of the nonlinear state $x_{t+1|t}^n$. This is performed in Step 4(b). Now, consider model (3) conditioned on the nonlinear state variable. The conditioning implies that (3a) can be thought of as a measurement equation. This is used in step 4(c) together with a time update of the linear state estimates.

The estimates, as expected means, of the state variables and their covariances are given below.

$$\hat{x}_{t|t}^n = \sum_{i=1}^N \tilde{q}_t^{(i)} \hat{x}_{t|t}^{n,(i)}, \quad (9a)$$

$$\hat{P}_{t|t}^n = \sum_{i=1}^N \tilde{q}_t^{(i)} \left(\left(\hat{x}_{t|t}^{n,(i)} - \hat{x}_{t|t}^n \right) \left(\hat{x}_{t|t}^{n,(i)} - \hat{x}_{t|t}^n \right)^T \right), \quad (9b)$$

$$\hat{x}_{t|t}^l = \sum_{i=1}^N \tilde{q}_t^{(i)} \hat{x}_{t|t}^{l,(i)}, \quad (9c)$$

$$\hat{P}_{t|t}^l = \sum_{i=1}^N \tilde{q}_t^{(i)} \left(P_{t|t}^{(i)} + \left(\hat{x}_{t|t}^{l,(i)} - \hat{x}_{t|t}^l \right) \left(\hat{x}_{t|t}^{l,(i)} - \hat{x}_{t|t}^l \right)^T \right), \quad (9d)$$

where $\{\tilde{q}_t^{(i)}\}_{i=1}^N$ are the normalized importance weights, provided by step 2 in Algorithm H.1.

3 Analysis

In this section, several properties important in the practical application of the marginalized particle filter are analyzed. First, the variance reduction inherent using the Rao-Blackwellization idea is explained. Second, the computational burden of MPF is analyzed in detail. Finally, quantization effects in the measurement relation are described.

3.1 Variance Reduction

The variance of a function or estimator $g(U, V)$, depending on two random variables, U and V can be written as

$$\text{Var}\{g(U, V)\} = \text{Var}\{\text{E}\{g(U, V)|V\}\} + \text{E}\{\text{Var}\{g(U, V)|V\}\}, \quad (10)$$

Hence, in principle, the conditional inequality

$$\text{Var}\{\text{E}\{g(x_t^l, X_t^n)|X_t^n\}\} \leq \text{Var}\{g(x_t^l, X_t^n)\}, \quad (11)$$

can be employed. This is sometimes referred to as Rao-Blackwellization, see, e.g., Robert and Casella (1999). This is the basic part that improves performance using the marginalization idea. In the MPF setup, U and V are represented by the linear and nonlinear states.

3.2 Computational Complexity

In discussing the use of the MPF it is sometimes better to partition the state vector into one part that is estimated using the particle filter $x_t^p \in \mathbf{R}^p$ and one part that is estimated using the Kalman filter $x_t^k \in \mathbf{R}^k$. Obviously all the nonlinear states x_t^n are included in x_t^p . However, we could also choose to include some of the linear states there as well. Under the assumption of linear dynamics, this notation allows us to write (3) according to

$$x_{t+1}^p = A_t^p x_t^p + A_t^k x_t^k + w_t^p, \quad w_t^p \sim \mathcal{N}(0, Q_t^p), \quad (12a)$$

$$x_{t+1}^k = F_t^p x_t^p + F_t^k x_t^k + w_t^k, \quad w_t^k \sim \mathcal{N}(0, Q_t^k), \quad (12b)$$

$$y_t = h_t(x_t^p) + C_t x_t^k + e_t, \quad e_t \sim \mathcal{N}(0, R_t). \quad (12c)$$

First, the case $C_t = 0$ is discussed. For instance, the first instruction $P_{t|t}(A_t^k)^T$ corresponds to multiplying $P_{t|t} \in \mathbf{R}^{k \times k}$ with $(A_t^k)^T \in \mathbf{R}^{k \times p}$, which requires pk^2 multiplications and $(k-1)kp$ additions (Golub and Van Loan, 1996). The total equivalent flop (EF)¹ complexity is derived by Karlsson et al. (2005),

$$\begin{aligned} \mathcal{C}(p, k, N) \approx & 4pk^2 + 8kp^2 + \frac{4}{3}p^3 + 5k^3 - 5kp + 2p^2 \\ & + (6kp + 4p^2 + 2k^2 + p - k + pc_3 + c_1 + c_2)N. \end{aligned} \quad (13)$$

Here, the coefficient c_1 has been used for the calculation of the Gaussian likelihood, c_2 for the resampling and c_3 for the random number complexity. Note that, when $C_t = 0$

¹The EF complexity for an operation is defined as the number of flops that result in the same computational time as the operation.

the same covariance matrix is used for all Kalman filters, which significantly reduce the computational complexity.

By requiring $\mathcal{C}(p+k, 0, N_{PF}) = \mathcal{C}(p, k, N(k))$, where N_{PF} corresponds to the number of particles used in the standard particle filter we can solve for $N(k)$. This gives the number of particles $N(k)$ that can be used by the MPF in order to obtain the same computational complexity as if the standard particle filter had been used for all states. In Figure 2 the ratio $N(k)/N_{PF}$ is plotted for systems with $m = 3, \dots, 9$ states. Hence, using

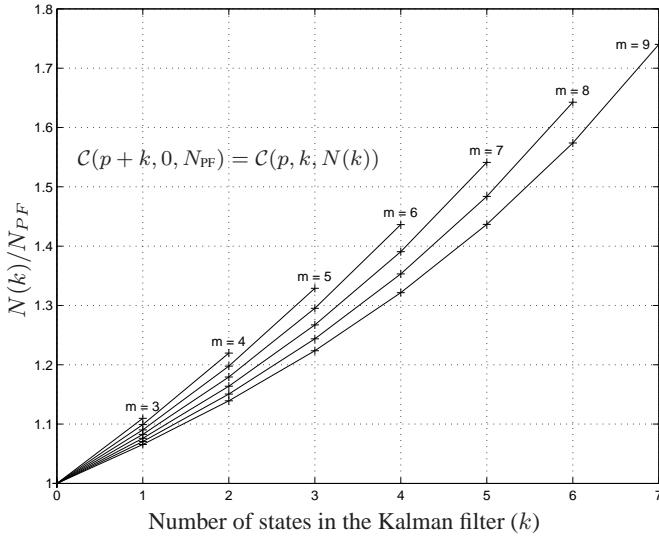


Figure 2: Ratio $N(k)/N_{PF}$ for systems with $m = 3, \dots, 9$ states and $C_t = 0, n = 2$ is shown. It is apparent the MPF can use more particles for a given computational complexity, when compared to the standard PF.

Figure 2 it is possible to directly find out how much there is to gain in using the MPF from a computational complexity point of view. The figure also shows that the computational complexity is always reduced when the MPF can be used instead of the standard particle filter. Furthermore, as previously mentioned, the quality of the estimates will improve or remain the same when the MPF is used (Doucet et al., 2001b).

Second, if $C_t \neq 0$, the Riccati recursions have to be evaluated separately for each particle. This results in a significantly increased computational complexity. Hence, different covariance matrices have to be used for each Kalman filter, implying that (13) has to be modified. Approximately the complexity is given by Karlsson et al. (2005),

$$\begin{aligned} \mathcal{C}(p, k, N) \approx & (6kp + 4p^2 + 2k^2 + p - k + pc_3 + c_1 + c_2 \\ & + 4pk^2 + 8kp^2 + \frac{4}{3}p^3 + 5k^3 - 5kp + 2p^2 + k^3)N. \end{aligned} \quad (14)$$

In Figure 3 the ratio $N(k)/N_{PF}$ is plotted for systems with $m = 3, \dots, 9$ states. For

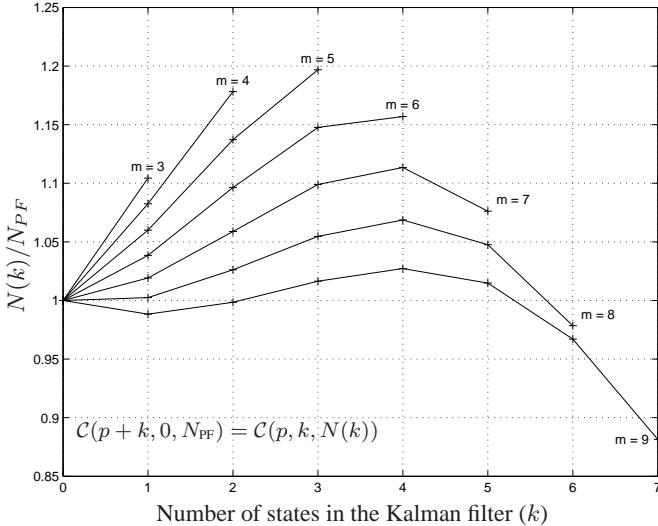


Figure 3: Ratio $N(k)/N_{PF}$ for systems with $m = 3, \dots, 9$ states and $C_t \neq 0, n = 2$ is shown. For systems with high state dimension and many marginalized states the standard PF can use more particles than the MPF.

systems with few states the MPF is more efficient than the standard particle filter. However, for systems with more states, where most of the states are marginalized the standard particle filter becomes more efficient than the MPF. This is due to the Riccati recursions mentioned above.

3.3 Quantization Effects

When implementing filters or estimators in hardware, the calculations can usually be performed with sufficient precision. However, the sensor or measurement relation may not always have sufficient resolution. This is referred as measurement quantization, and is a common problem in for instance telecommunication, where the channel bandwidth is limited. To be able to use limited communication resources, severe quantization may be needed. Also for large sensor networks applications, many very simple and cheap sensors with large quantization effects are used. Furthermore, many sensors or signal processing devices are naturally quantized, for instance range measurements in a pulsed radar or pixelized information from a vision system.

Here we will discuss quantization using a multi-level uniform quantization. Consider the problem of estimating x from the quantized measurements $y = \mathcal{Q}_m(x + e)$. The uniform quantization discussed here is implemented as the *midriser* quantizer, as described

in Lipshitz et al. (1992). If not saturated it is given as

$$\mathcal{Q}_m(z) = \Delta \left\lfloor \frac{z}{\Delta} \right\rfloor + \frac{\Delta}{2}. \quad (15)$$

Here, $\mathcal{Q}_m(\cdot)$ denotes the nonlinear quantization mapping with m levels, all with equal quantization height Δ . The $\lfloor \cdot \rfloor$ operator rounds downwards to the nearest integer. To keep a unified notation with the sign quantization $\mathcal{Q}_1(z) = \text{sign}(z)$, the midriser convention will be used, so $y \in \{-m\Delta + \frac{\Delta}{2}, \dots, (m-1)\Delta + \frac{\Delta}{2}\}$, with $\Delta = 2^{-b}$, using b bits, $2m = 2^b$ levels and $2^b - 1$ thresholds. The sign quantization corresponds to $b = 1$, $m = 1$ and $\Delta = 2$ in this notation.

In Karlsson and Gustafsson (2005b), this static problem is analyzed using the *maximum likelihood* (ML) estimator. The performance is also investigated using the Fisher information or *Cramér-Rao lower bound* (CRLB). The resulting likelihood function can also be used in the particle filter, allowing for a statistically correct treatment of measurement quantization effects in dynamic systems. If the model is in accordance with the requirement of the MPF algorithm, it is possible to handle the nonlinearity introduced by the quantization in the measurement equation in the MPF. In Karlsson and Gustafsson (2005b) different quantizers are studied. Below, only the simplest sign quantizer, $y_t = \mathcal{Q}_1(x_t + e_t)$, $e_t \sim \mathcal{N}(0, \sigma^2)$, is discussed. The probability function for y can be calculated using

$$p(y = -1|x) = \Pr(x + e < 0) = \Pr(e < -x) = \int_{-\infty}^{-x} \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{t^2}{2\sigma^2}} dt \quad (16)$$

$$= \int_{-\infty}^{-x/\sigma} \frac{1}{\sqrt{2\pi}} \exp^{-\frac{t^2}{2}} dt \stackrel{\Delta}{=} \Phi(-x/\sigma). \quad (17)$$

Similarly,

$$p(y = +1|x) = \Pr(x + e \geq 0) = 1 - \Phi(-x/\sigma). \quad (18)$$

Hence, the discrete likelihood needed in the PF/MPF, in (5), can be written as

$$p(y|x) = \Phi(-x/\sigma) \delta(y+1) + (1 - \Phi(-x/\sigma)) \delta(y-1), \quad (19)$$

where

$$\delta(i) = \begin{cases} 1, & i = 0, \\ 0, & i \neq 0. \end{cases} \quad (20)$$

The calculated likelihood can be used in the PF/MPF to incorporate the quantization effect in a statistically correct way. Similar for multi-level quantization.

Example H.1: Filtering – sign quantizer

Consider the following scalar system with a sign quantizer

$$x_{t+1} = F_t x_t + w_t, \quad x_0 = 0, \quad (21a)$$

$$y_t = \mathcal{Q}_1(x_t + e_t), \quad (21b)$$

where

$$F_t = 0.95, \quad \text{Var } w_t = 0.10^2, \quad \text{Var } e_t = 0.58^2. \quad (22)$$

In Figure 4 the RMSE for the KF and the PF are presented using 200 Monte Carlo simulations. The measurement noise in the KF was adjusted in the filter to handle the quantization by adding an extra variance of $\Delta^2/12$. The PF used the correct sign quantized likelihood using 1000 particles. The theoretical Cramér-Rao lower bound is also given in Figure 4. For details, see Karlsson and Gustafsson (2005b).

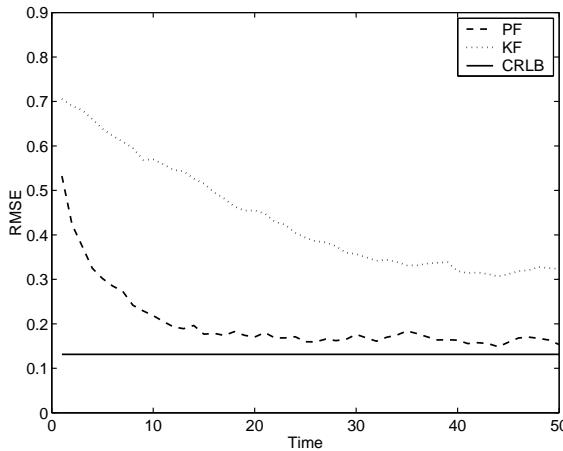


Figure 4: RMSE for the PF and KF for a linear Gaussian system with a sign quantizer in the measurement relation, compared to the Cramér-Rao lower bound.

Note that for the example presented only one state was used, hence no marginalization was applied. If the problem is formulated with linear, Gaussian dynamics and quantization in the measurement, these nonlinear states can be handled by the PF and the rest by the KF in the MPF framework.

4 Introducing the Applications

As discussed in the previous section, the different estimation methods handle nonlinearities in different ways. In the applications studied in this paper a framework consisting of linear, Gaussian system dynamics and nonlinear measurements is considered. Basically, two different areas are studied: *GPS-free positioning*, where the aim is to estimate the own platform's position and *target tracking*, where the state of an unknown, observed target is estimated from measurements. These applications also represent typical examples where sensor fusion techniques are important. The MPF provides an efficient way to incorporate both linear and nonlinear measurement relations. Both results from simulated data and experimental data are presented. More precisely, the studied applications are:

Positioning applications:

- *Underwater terrain-aided positioning:* Depth information from a geographical information system (GIS) database is used together with sonar depth measurements to improve positioning. A demonstrator system has been developed in co-operation with Saab Underwater Systems.
- *Aircraft terrain-aided positioning:* A height GIS database is used together with radar height measurements to improve the position, compared to only inertial navigation system (INS) measurements. A demonstrator system has been developed by Saab Aerospace.
- *Automotive map-aided positioning:* Utilizing wheel speed sensors from the ABS and information from a street-map database, car positioning independent of GPS is possible. This is available as a commercial product from NIRA Dynamics.

Target tracking applications:

- *Automotive target tracking:* Intelligent automotive systems require information about the host vehicle and its surroundings (road geometry and the position of surrounding vehicles). Using vision and radar measurements, the corresponding estimation problem is addressed. A demonstrator vehicle has been developed in co-operation with Volvo Car Corporation.
- *Bearings-only target tracking:* When passive sensors, such as an infrared (IR) sensor are used, we can only measure the direction, bearing, to the unknown target. However, by appropriate maneuvering, the range and range rate can be estimated. This is studied in an air-to-sea application, i.e., an aircraft tracking a ship.
- *Radar target tracking:* A radar sensor measures at least range and direction (azimuth, elevation) to the target. In this particular application the computational aspects of the MPF are studied in detail.

The dynamic models employed in the applications all have a linear motion model and a nonlinear measurement model. By partitioning the state vector x_t into two parts, one for the linear state variables x_t^l and one for the nonlinear state variables x_t^n the model fits the MPF framework perfectly. For example, consider Cartesian position coordinates (X, Y, Z) and introduce the state vector $x_t = (X_t, Y_t, Z_t, \dot{X}_t, \dot{Y}_t, \dot{Z}_t)^T$, with position and velocity states. In target tracking the relative distance between the target and the observation platform is often used as state. Furthermore, the first-order derivatives of this distance, relative velocity, are included in the state vector. The resulting motion model is given by

$$x_{t+1} = F_t x_t + G_t w_t, \quad (23a)$$

where

$$F_t = \begin{pmatrix} I_3 & TI_3 \\ O_3 & I_3 \end{pmatrix}, \quad G_t = \begin{pmatrix} \frac{T^2}{2} I_3 \\ TI_3 \end{pmatrix}, \quad (23b)$$

Here, I_3 denotes the 3×3 unity matrix and O_3 denotes the 3×3 null matrix. The measurement relation is in the sequel treated as a nonlinear relation of the state, subject to additive measurement noise,

$$y_t = h(x_t^n) + e_t. \quad (23c)$$

It can for instance represent range and bearing measurement from a radar, height or depth measurements for terrain navigation applications. In all these situations it is a function of the position states. For the example above, $x_t^n = (X_t, Y_t, Z_t)^T$ and $x_t^l = (\dot{X}_t, \dot{Y}_t, \dot{Z}_t)^T$. Another common state variable is the heading or course.

For a more thorough discussion regarding models for positioning, navigation, and tracking applications within the present setting, see Gustafsson et al. (2002). Interesting to note is also that common phenomena such as bias or scale-factor estimation can often be introduced in the linear, Gaussian sub-system. Hence, the MPF provides an efficient way to handle such problems.

5 Positioning Applications

This section is concerned with position estimation, where information from geographical information systems is used together with different *distance measurement equipment* (DME). First, an underwater positioning method based on sonar depth measurements is presented. Second, the same idea is employed to solve the aircraft positioning problem using height measurements from a radar altimeter. Finally, the automotive positioning problem is briefly presented.

5.1 Underwater Terrain-aided Positioning

In this section we describe an *underwater* positioning method based DME information from sonar depth readings and a comparison with a depth database to find the position of the host vessel. It is based on the preliminary studies in Karlsson et al. (2003), Karlsson and Gustafsson (2003), together with Karlsson and Gustafsson (2005a).

Using a sonar sensor and a *differential GPS* (DGPS), an underwater depth map was constructed, illustrated in Figure 5, together with the platform at depth $d_t = 0$ and with sonar depth measurements r_t . After the data for map generation was collected, an independent test run in the map region was performed, in order to collect measurements to test the PF/MPF map-aided positioning system. In Karlsson and Gustafsson (2003) a coordinated turn model extended with bias terms was used. In order to apply the MPF a Taylor expansion was calculated, enabling for a model approximately in the correct form. The estimation performance reported for the MPF was similar to the PF, but to a much smaller computational burden. In order to fit the linear, Gaussian dynamics framework, we will only consider the model from Karlsson and Gustafsson (2005a). The number of particles used initially was $N = 50000$, but quickly reduced to $N = 10000$, when the particle cloud had most of its particles clustered around the true position. The result is presented in Figure 6, where the parametric CRLB is calculated using an extended Kalman filter, evaluated around the true trajectory.

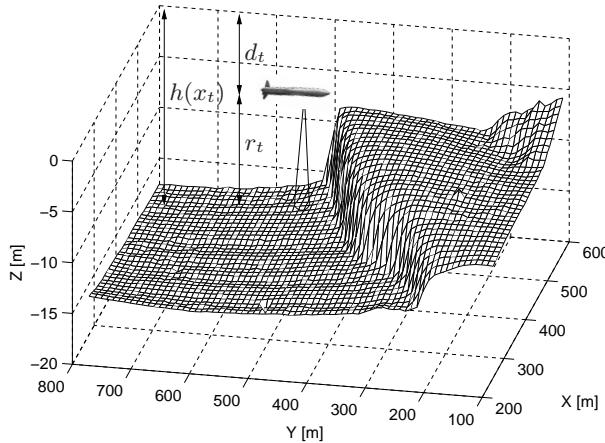


Figure 5: Underwater positioning using sonar depth measurements and a true terrain database. The sonar depth is d_t , and the sonar indicates the relative range to the sea floor r_t . The database gives $h(x_t)$.

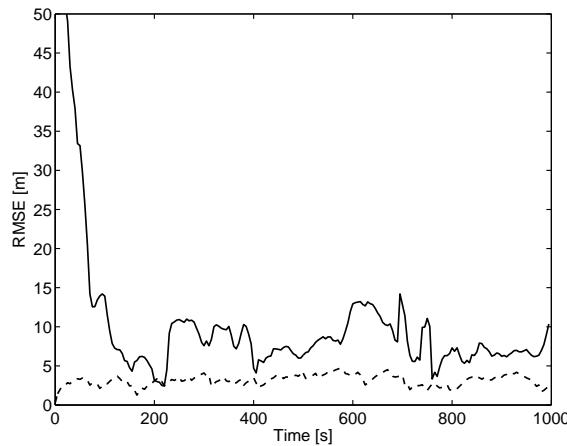


Figure 6: The position RMSE from the PF (solid line) using the experimental test data together with the parametric CRLB (dashed line) as the EKF solution around the true trajectory. The nominal speed is between 0.9 – 1.5 m/s. Note that only one experimental test run was available for the RMSE calculation.

5.2 Aircraft Terrain-Aided Positioning

The Swedish fighter aircraft Gripen is equipped with an accurate radar altimeter as DME sensor and a terrain elevation database, similar to the discussion in the previous section. These measurements are used together with an *inertial navigation system* (INS) in order to solve the aircraft positioning problem. This problem has previously been studied, see,

e.g., Bergman (1999), Svensson (1999), Ahlström and Calais (2000). The overall structure of the model used in this application is in the form (12), with the following measurement equation,

$$y_t = h \left(\begin{pmatrix} X_t \\ Y_t \end{pmatrix} + x_t^n \right) + e_t, \quad (24)$$

where X_t and Y_t denotes the error in latitude and longitude respectively. The feasibility study performed used a sub-model with nine states. This sub-model contains all ingredients of the total system and the principle is scalable to the full model with 27 states. For details regarding the model we refer to Nordlund (2002) and the references therein.

The measurement equation (24) is highly nonlinear, due to the use of the terrain elevation database. This implies that the EKF cannot be used. Furthermore, the high dimension of the problem prevents the use of the particle filter. However, the model structure fits perfectly into the marginalized particle filter framework. This approach has been evaluated using authentic flight data with promising results, see Figure 7 where we provide a plot of the error in horizontal position for a different number of particles. From this plot it is clear

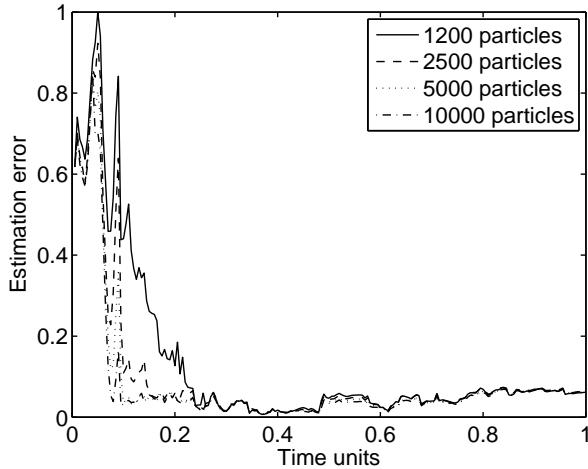


Figure 7: Horizontal position error as a function of time units for different numbers of particles. Note that the scale has been normalized for confidentiality reasons.

that the main difference in performance is in the transient phase, in the stationary phase the performance is less sensitive to the number of particles used. Hence, the idea of using more particles in the transient phase suggests itself. This idea was used, for the same reason, in the previous section as well. For a more detailed account on these experiments, see Frykman (2003), Schön et al. (2005), Nordlund (2002).

5.3 Automotive Map-Aided Positioning

The idea is to use the information available from the wheel speed sensors together with digital map information to estimate the position of the car, without the need for GPS information. The resulting problem is nonlinear and fits into the framework provided by the particle filter and the marginalized particle filter. For further details on this approach, see, e.g., Forssell et al. (2002), Hall (2000), Svenzén (2002).

6 Target Tracking Applications

In this section three target tracking applications are studied. First, an automotive target tracking problem is discussed. This is followed by a bearings-only estimation problem. Finally, a radar target tracking application highlight different computational aspects of the marginalized particle filter.

6.1 Automotive Target Tracking

This application deals with the problem of estimating the vehicle surroundings (road geometry and the position of other vehicles), which is required by advanced automotive safety systems, such as adaptive cruise control, collision avoidance and lane guidance. For a thorough treatment of this application, see Eidehall et al. (2005).

The main difference between tracking in automotive applications and tracking in other applications, such as air traffic control or naval tracking, is that in automotive tracking it can be assumed that the motion of the tracked objects, with a certain probability, is constrained to the road. In order to be able to use and benefit from this fact we make use of a curved coordinate system which is attached to and follows the road (Eidehall, 2004). The measurements are provided by a vision system and a radar system. The vision system provides measurements of the road curvature, the yaw angle and the distance to the right and left lane markings. Furthermore, the radar provides range measurements to the surrounding vehicles. The final model, thoroughly derived in Eidehall (2004), Eidehall et al. (2005) is in the form (12), which opens up for using the marginalized particle filter. The nonlinear part of the measurement equation for a given target i is

$$y_t = h(X_t^i, Y_t^i) + e_t, \quad (25)$$

where $h(\cdot)$ described the geometric transformation from a curved, road-aligned coordinate system to a Cartesian coordinate system, in which the measurements are registered. For details, see Eidehall et al. (2005). In evaluating the estimation performance we study the estimate of the road curvature. It is crucial to several automotive applications, such as adaptive cruise control systems, collision warning or any system that relies on assigning leading vehicles to the correct lane. For a leading vehicle 100 m in front of the host vehicle, a small curvature error of, say $0.5 \cdot 10^{-3} \text{ m}^{-1}$ will result in an error of 2.5 m in the lateral direction (Eidehall et al., 2005). This is enough to assign the leading vehicle to the wrong lane.

The data set used was collected in the northern parts of Sweden during winter. This implies that the visibility is low, which in turn implies that the measurements from the

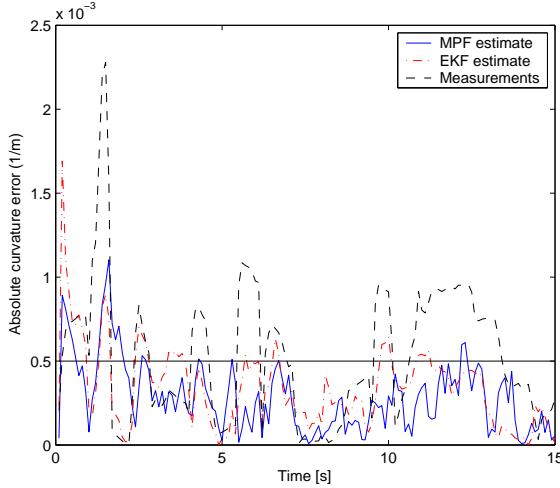


Figure 8: The absolute curvature error. Here, we have indicated the level $0.5 \cdot 10^{-3} \text{ m}^{-1}$, which is used as a motivating example in the text. For errors above this level, leading vehicles at a distance of 100 m are likely to be assigned to the wrong lane.

vision system definitely have to be supported by the radar measurements to obtain a solid overall estimate. In Figure 8 we provide the absolute curvature estimation error using the MPF and the EKF. Furthermore, the raw vision measurement of the curvature is also included. From Figure 8 it is clear that both filters improve the quality of the curvature estimate substantially. However, the performance of the MPF is only slightly better than the EKF. Hence, in this particular setting it might be hard to motivate using the MPF, due to its higher computational complexity. If we were to use more advanced measurement equations, such as those based on map information the MPF might be the only option.

6.2 Bearings-Only Target Tracking

In this section, an air-to-sea bearings-only application is studied. Assume that the ship (target) and the aircraft (tracking platform) are described by the same type of linear dynamics as in Section 4 for the position and the velocity, save for the fact the relative quantities have been used as states. For bearings-only applications the measurement relation for the azimuth angle φ and elevation angle θ is given as

$$y_t = h(x_t) + e_t = \begin{pmatrix} \varphi_t \\ \theta_t \end{pmatrix} + e_t = \begin{pmatrix} \arctan(Y_t/X_t) \\ \arctan\left(\frac{-Z_t}{\sqrt{X_t^2+Y_t^2}}\right) \end{pmatrix} + e_t, \quad (26)$$

where X_t , Y_t , and Z_t denote the Cartesian components of the relative position.

In a simulation study the range estimation problem using an infrared (IR) sensor is considered. The PF and MPF are compared to a bank of EKFs, using the *range parameter*

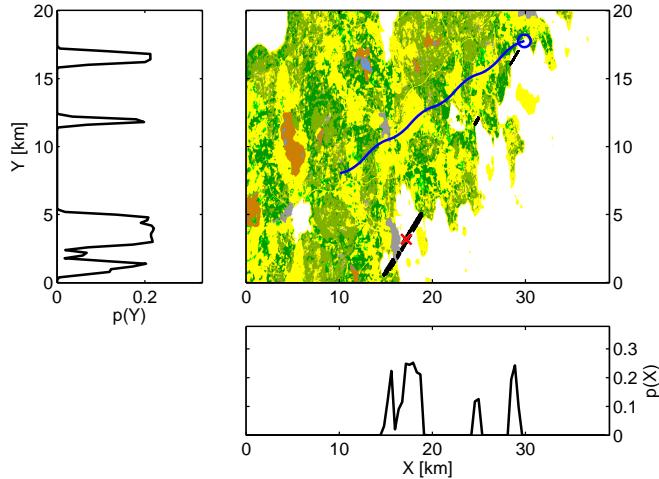


Figure 9: The position for the aircraft (\circ) and the target ship (\times) together with the marginalized position pdf using the particle filter with terrain induced constraints at $t = 1$ s. The particle cloud and the future trajectory of the aircraft are also shown.

terized extended Kalman filter (RPEKF) method, (Peach, 1995, Arulampalam and Ristic, 2000). The relative distance and the aircraft trajectory are illustrated in Figure 9. The target model used in the simulations assumes a small constant velocity. The terrain database has a resolution of 50 m. In Figure 9 the scenario is presented together with the marginal position densities in each direction, $p(X)$ and $p(Y)$, for time $t = 1$ s, using terrain constraints. In Figure 10 the position RMSE is presented for the PF and the MPF with and without the map constraints, and for the RPEKF. Obviously the incorporation of constraints improves the performance. The different particle filters have basically the same performance for this scenario. For details regarding the simulation study, see Karlsson and Gustafsson (2005c), where similar bearings-only applications are described in detail, both for simulated data and for experimental data. For instance, experimental data from a passive sonar system on a torpedo is used for bearings-only tracking.

6.3 Radar Target Tracking

In this section, the radar target tracking application from Karlsson et al. (2005) is highlighted. The general method for analyzing the computational complexity presented in Karlsson et al. (2005) and briefly reviewed in Section 3.2, is illustrated using a common target tracking model. The problem of estimating the position and velocity of an aircraft is studied using the dynamics from Section 4, and the following measurement equation, which gives the range and azimuth from the radar system,

$$y_t = h(x_t) + e_t = \begin{pmatrix} \sqrt{X_t^2 + Y_t^2} \\ \arctan(Y_t/X_t) \end{pmatrix} + e_t, \quad (27)$$

where $\text{Cov}\{w\} = \text{diag}(1 \ 1 \ 1 \ 1 \ 0.01 \ 0.01)$, $\text{Cov}\{e\} = \text{diag}(100 \cdot 10^{-6})$ and the state vector is $x_t = (X \ Y \ \dot{X} \ \dot{Y} \ \ddot{X} \ \ddot{Y})^T$, i.e., position, velocity and acceleration.

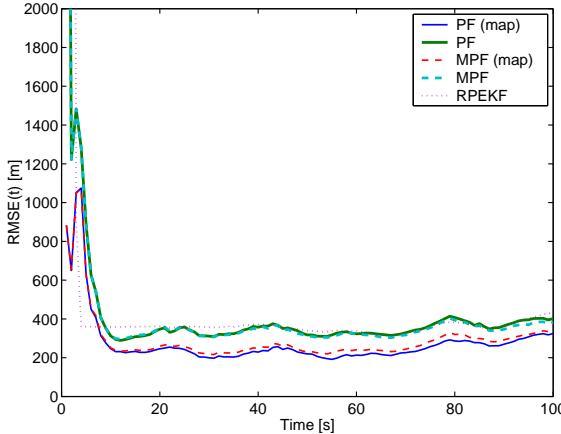


Figure 10: Position $\text{RMSE}(t)$ for air-to-sea passive ranging using 100 Monte Carlo simulations.

The model has two nonlinear state variables and four linear state variables. Two cases are now studied, the full PF, where all states are estimated using the PF and the completely marginalized PF, where all linear states are marginalized out and estimated using the KF. If we want to compare the two approaches under the assumption that they use the same computational resources, i.e., $\mathcal{C}(6, 0, N_{\text{PF}}) = \mathcal{C}(2, 4, N_{\text{MPF}})$, we obtain

$$N_{\text{PF}} = \underbrace{\left(1 - \frac{4c_3 + 56}{c_1 + c_2 + 6c_3 + 150}\right)}_{<1} N_{\text{MPF}}. \quad (28)$$

From (28) it is clear that for a given computational complexity more particles can be used in the MPF than in the standard PF. This is verified experimentally in Karlsson et al. (2005).

Using a constant computational complexity the number of particles that can be used is computed. The study is performed by first running the full PF and measure the time consumed by the algorithm. An Monte Carlo simulation, using $N = 2000$ particles, is performed in order to obtain a stable estimate of the time consumed by the algorithm. In Table 6.3 the number of particles (N), the total RMSE from 100 Monte Carlo simulations, and the simulation times are shown for the different marginalization cases. From Table 6.3 it is clear that the different MPFs can use more particles for a given time, which is in perfect correspondence with the theoretical result given in (28).

Let us now discuss what happens if a constant velocity RMSE is used. First the velocity RMSE for the full PF is found using an Monte Carlo simulation. This value is then used as a target function in the search for the number of particles needed by the different MPFs. Table 2 clearly indicates that the MPF can obtain the same RMSE using fewer particles. The result is that using full marginalization only requires 14% of the computational resources as compared to the standard PF in this example.

Table 1: Results from the simulation, using a constant computational complexity. If a certain state variable is estimated using the PF this is indicated with a P and if the KF is used this is indicated with a K.

	PPPPP	PPKKP	PPPKK	PPKKKK
<i>N</i>	2000	2029	1974	2574
RMSE pos	7.10	5.81	5.76	5.60
RMSE vel	3.62	3.27	3.28	3.21
RMSE acc	0.52	0.47	0.45	0.44
Time	0.59	0.58	0.57	0.60

Table 2: Results using a constant velocity RMSE.

	PPPPPP	PPKKP	PPPKK	PPKKKK
<i>N</i>	2393	864	943	264
RMSE pos	7.07	6.98	7.12	7.27
RMSE vel	3.58	3.60	3.65	3.61
RMSE acc	0.50	0.51	0.49	0.48
Time	0.73	0.26	0.28	0.10

7 Concluding Discussion

In this paper several positioning and target tracking applications are solved using the marginalized particle filter. In the framework employed the dynamic motion models are linear, subject to Gaussian noise and the measurement models are nonlinear. This important special case of the general MPF allows for an efficient implementation.

The computational complexity of the MPF algorithm is thoroughly analyzed for a radar application, but because of the similarities in the studied models in the applications, these results are approximately valid for them as well. The radar application also illustrates another important property of the MPF, namely that the quality of the estimates is enhanced compared to the standard particle filter.

Another unifying feature among the various applications is that they all use measurements from various different sources, implying that we are indeed solving the *sensor fusion* problem using the MPF. Terrain-aided positioning problems are quite hard to handle using methods based on linearization, due to the fact that it is very hard to obtain a good linear description of the map database, used to form the measurement equations. Hence, the MPF is a very powerful tool for these applications. We saw that the computational complexity can be reduced substantially by decreasing the number of particles when the stationary phase is reached. This is a common idea, employed in all the applications, since more computational resources should be used in the transient phase.

Common for the measurement relation is that nonlinearities and non-Gaussian noise is handled in a statistically optimal way, by the particle filter. Particularly, if the measurement relation is subject to severe quantization this is important to handle. Quantization arises naturally in many applications, but typically in sensor networks where sensor fusion is applied based on information from a large number of very cheap sensors, this can be a major issue.

Acknowledgment

This work was supported by VINNOVA's Center of Excellence ISIS (Information Systems for Industrial Control and Supervision), and by the Swedish Research Council (VR).

References

- Ahlström, M. and Calais, M. (2000). Bayesian terrain navigation with monte carlo method. Master's Thesis No LiTH-ISY-EX-3051, Department of Electrical Engineering, Linköping University, Sweden.
- Anderson, B. D. O. and Moore, J. B. (1979). *Optimal Filtering*. Information and system science series. Prentice Hall, Englewood Cliffs, NJ, USA.
- Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society*, 64(4):827–836.
- Andrieu, C. and Godsill, S. J. (2000). A particle filter for model based audio source separation. In *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000)*, Helsinki, Finland.
- Arulampalam, S. and Ristic, B. (2000). Comparison of the particle filter with range-parameterized and modified polar EKFs for angle-only tracking. In *Proceedings of SPIE, Signal and Data Processing of Small Target*, pages 288–299, Orlando, FL, USA.
- Bergman, N. (1999). *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Dissertations No 579, Linköping Studies in Science and Technology, SE-581 83 Linköping, Sweden.
- Casella, G. and Robert, C. P. (1996). Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94.
- Chen, R. and Liu, J. S. (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society*, 62(3):493–508.
- Chen, R., Wang, X., and Liu, J. S. (2000). Adaptive joint detection in flat-fading channels via mixture Kalman filtering. *IEEE Transactions on Information Theory*, 46(6):2079–2094.
- Daly, M. J., Reilly, J. P., and Morelande, M. R. (2005). Rao-Blackwellised particle filtering for blind system identification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, USA.
- Doucet, A., de Freitas, N., and Gordon, N., editors (2001a). *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, USA.
- Doucet, A., Godsill, S. J., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.

- Doucet, A., Gordon, N., and Krishnamurthy, V. (1999). Particle filters for state estimation of jump Markov linear systems. Technical Report CUED/F-INFENG/TR 359, Signal Processing Group, Department of Engineering, University of Cambridge, Trupington street, CB2 1PZ Cambridge.
- Doucet, A., Gordon, N., and Krishnamurthy, V. (2001b). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624.
- Eidehall, A. (2004). *An Automotive Lane Guidance System*. Licentiate Thesis No 1122, Department of Electrical Engineering, Linköping University, Sweden.
- Eidehall, A., Schön, T. B., and Gustafsson, F. (2005). The marginalized particle filter for automotive tracking applications. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 369–374, Las Vegas, USA.
- Forssell, U., Hall, P., Ahlqvist, S., and Gustafsson, F. (2002). Novel map-aided positioning system. In *Proceedings of FISITA*, Helsinki, Finland.
- Frykman, P. (2003). Applied particle filters in integrated aircraft navigation. Master's Thesis No LiTH-ISY-EX-3406, Department of Electrical Engineering, Linköping University, Sweden.
- Giremus, A. and Tourneret, J. Y. (2005). Joint detection/estimation of multipath effects for the global positioning system. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 17–20, Philadelphia, PA, USA.
- Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*. John Hopkins University Press, Baltimore, third edition.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113.
- Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J. (2002). Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437.
- Hall, P. (2000). A Bayesian approach to map-aided vehicle positioning. Master's Thesis No LiTH-ISY-EX-3102, Department of Electrical Engineering, Linköping University, Sweden.
- Kailath, T., Sayed, A. H., and Hassibi, B. (2000). *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, NJ, USA.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45.
- Karlsson, R. and Gustafsson, F. (2003). Particle filter for underwater navigation. In *Proceedings of the Statistical Signal Processing Workshop*, pages 509–512, St. Louis, MO, USA.

- Karlsson, R. and Gustafsson, F. (2005a). Bayesian surface and underwater navigation. Technical Report LiTH-ISY-R-2674, Department of Electrical Engineering. Submitted to IEEE Transactions on signal processing.
- Karlsson, R. and Gustafsson, F. (2005b). Particle filtering for quantized sensor information. In *Proceedings of the 13th European Signal Processing Conference*, Antalya, Turkey.
- Karlsson, R. and Gustafsson, F. (2005c). Recursive Bayesian estimation – bearings-only applications. *IEE Proceedings on Radar, Sonar, and Navigation. Special issue on target tracking: Algorithms and Applications*, 152(5):305–313.
- Karlsson, R., Gustafsson, F., and Karlsson, T. (2003). Particle filter and Cramér-Rao lower bound for underwater navigation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong.
- Karlsson, R., Schön, T., and Gustafsson, F. (2005). Complexity analysis of the marginalized particle filter. *IEEE Transactions on Signal Processing*, 53(11):4408–4411.
- Li, P., Goodall, R., and Kadirkamanathan, V. (2003). Parameter estimation of railway vehicle dynamic model using Rao-Blackwellised particle filter. In *Proceedings of the European Control Conference*, Cambridge, UK.
- Lipshitz, S. P., Wannamker, R. A., and Vanderkooy, J. (1992). Quantization and dither: A theoretical survey. *Journal of Audio Engineering Society*, 40(5):355–375.
- Nordlund, P.-J. (2002). *Sequential Monte Carlo Filters and Integrated Navigation*. Licentiate Thesis No 945, Department of Electrical Engineering, Linköping University, Sweden.
- Peach, N. (1995). Bearings-only tracking using a set of range-parameterized extended Kalman filters. In *IEE Proceedings of Control Theory and Applications*, volume 142, pages 73–80.
- Ristic, B., Arulampalam, S., and Gordon, N. (2004). *Beyond the Kalman Filter: particle filters for tracking applications*. Artech House, London, UK.
- Robert, C. P. and Casella, G. (1999). *Monte Carlo Statistical Methods*. Springer texts in statistics. Springer, New York.
- Schön, T. and Gustafsson, F. (2003). Particle filters for system identification of state-space models linear in either parameters or states. In *Proceedings of the 13th IFAC Symposium on System Identification*, pages 1287–1292, Rotterdam, The Netherlands. Invited paper.
- Schön, T., Gustafsson, F., and Nordlund, P.-J. (2005). Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289.
- Sorenson, H. W. and Alspach, D. L. (1971). Recursive Bayesian estimation using Gaussian sum. *Automatica*, 7:465–479.

- Svensson, M. (1999). Implementation and validation of Bayesian terrain navigation. Master's Thesis No LiTH-ISY-EX-2039, Department of Electrical Engineering, Linköping University, Sweden.
- Svenzén, N. (2002). Real time implementation of map aided positioning using a Bayesian approach. Master's Thesis No LiTH-ISY-EX-3297-2002, Department of Electrical Engineering, Linköping University, Sweden.
- Wang, X., R. Chen, R., and Guo, D. (2002). Delayed-pilot sampling for mixture Kalman filter with application in fading channels. *IEEE Transactions on Signal Processing*, 50(2):241–254.

Paper I

Lane Departure Detection for Improved Road Geometry Estimation

Authors: Thomas B. Schön, Andreas Eidehall, and Fredrik Gustafsson.

Edited version of Technical Report LiTH-ISY-R-2714, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

The paper has been submitted to *IEEE Intelligent Vehicle Symposium*, Tokyo, Japan, June 2006.

Lane Departure Detection for Improved Road Geometry Estimation

Thomas B. Schön*, Andreas Eidehall*†, and Fredrik Gustafsson*

*Department of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden.
E-mail: (schon, eidehall, fredrik)@isy.liu.se.

†Vehicle Dynamics and Active Safety
Volvo Car Corporation
SE-405 31 Göteborg, Sweden
E-mail: acidehal@volvocars.com

Abstract

An essential part of future collision avoidance systems is to be able to predict road curvature. This can be based on vision data, but the lateral movement of leading vehicles can also be used to support road geometry estimation. This paper presents a method for detecting lane departures, including lane changes, of leading vehicles. This information is used to adapt the dynamic models used in the estimation algorithm in order to accommodate for the fact that a lane departure is in progress. The goal is to improve the accuracy of the road geometry estimates, which is affected by the motion of leading vehicles. The significantly improved performance is demonstrated using sensor data from authentic traffic environments.

Keywords: Automotive tracking, change detection, state estimation, Kalman filter, CUSUM algorithm.

1 Introduction

THIS paper is concerned with the problem of simultaneously estimating the position of surrounding vehicles and the road geometry. The position of the surrounding vehicles is measured using a vision system and a radar, whereas the shape of the road is measured using vision only. It has been shown that integrating the tracking of other vehicles with the tracking of the road geometry parameters can give better performance than

treating these problems separately (General, 2000, Dellaert and Thorpe, 1997, Eidehall and Gustafsson, 2004, Zomotor and Franke, 1997). A fundamental assumption is that leading vehicles will keep following their lane, and their lateral movement can thus be used to support the otherwise difficult process of road geometry estimation. For example, when entering a curve as in Figure 1 it can be seen that the vehicles ahead all start moving to the right and thus there is a high probability that the road is turning to the right. This information can be used to significantly improve the rather crude road geometry estimates provided by the vision system. The assumption introduced above can mathematically be



Figure 1: When entering a curve, all vehicles start moving in the lateral direction. This information can be used to support the road geometry estimate.

represented as $\dot{y}^i = 0$, where y^i is the lateral position of vehicle i . Note that y^i is the position in relation to the lane, not the position in global Cartesian coordinates or coordinates attached to the host vehicle. In order to efficiently handle this, a road aligned, curved coordinate system is employed. It is important to note that the assumption of zero lateral velocity of the leading vehicles does not hold when they depart from the lane. This is typically accounted for in the model by adding white noise to the equation. The amount of noise, parameterized by the covariance matrix Q_{lat} , that should be used is a compromise. On the one hand it needs to be small enough for the lateral movement of the tracked vehicles to in fact improve the road prediction. On the other hand, it needs to be large enough so that a lane departure of a leading vehicle is not misinterpreted as a curve entry. This exemplifies the fundamental compromise present in all recursive estimation problems, the trade-off between noise attenuation and tracking accuracy. This compromise is illustrated in Figure 2, where the estimated road curvature, one of the road geometry parameters, using two different filters is plotted; one filter with a high value of Q_{lat} and one filter with a low. For reasons of comparison, the true values for the road curvature which is obtained using the method proposed in Eidehall and Gustafsson (2006) and the raw measurements from the vision system are also included. It is interesting to compare the raw vision measurements to the result from the filter. This clearly illustrates the power of a model based sensor fusion approach.

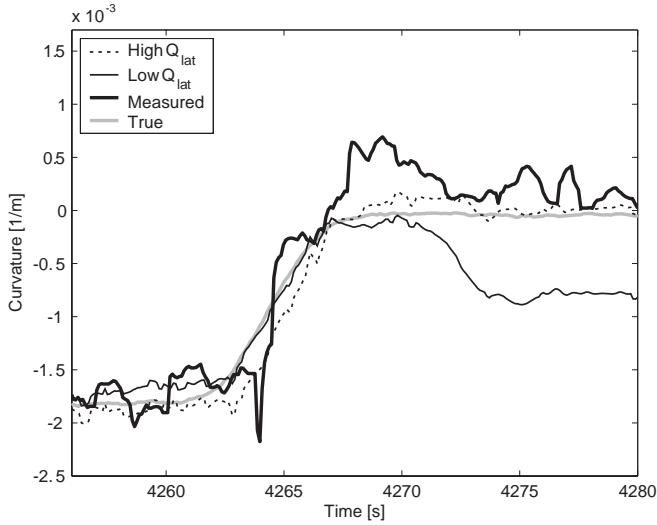


Figure 2: Comparison of estimation performance from two filters, one with a large Q_{lat} and one with a small Q_{lat} . The raw measurement signal from the image processing unit and the reference signal are also included. Comparing the raw vision measurement to the result from the filters clearly illustrates the power of a model based sensor fusion approach.

In Figure 2, an exit phase of a curve where the curvature suddenly drops from about $1.8 \cdot 10^{-3} \text{ m}^{-1}$ to zero can be seen. In this particular scenario there are two leading vehicles that can support the curvature estimate, see Figure 1. It can be seen that the filter with a low value of Q_{lat} performs much better during the curve exit and this is how we would really like to tune our filter. However, at a later stage the performance of this filter deteriorates. If the recorded video is studied, see Figure 3, it can be seen that this performance degradation coincides exactly with a lane change of one of the leading vehicles. The filter with a higher value of Q_{lat} does not suffer from this problem, but on the other hand it has a time delay in the estimate during the curve exit.

The aim of this paper is to detect lane departures of the leading vehicles and adapt the models accordingly, in order to obtain an improved road geometry estimate. When the lane departures have been detected, the compromise discussed above can systematically be resolved. This is accomplished by using a small Q_{lat} when the assumption $\dot{y}_i = 0$ is valid and only increase Q_{lat} during lane departure maneuvers.

Detection of lane departures and other model changes in automotive tracking has previously been studied, for example in Kaempchen et al. (2004), Weiss et al. (2004), where Interacting Multiple Models (IMM) (Bar-Shalom and Li, 1993) are used. However, their purpose is to improve the position estimates of the surrounding objects, rather than the road geometry parameters. Another approach is presented in Yen et al. (2004), where a neural network is used to detect lateral movement in a vision based system. The method we propose is different and based on the standard *cumulative sum* (CUSUM) algorithm



Figure 3: A snapshot from the video just after time 4270 s, where the lane change of the tracked vehicle commences.

(Page, 1954, Gustafsson, 2000), which is augmented with a module for correcting the error caused by using the wrong model during the detection phase, before the CUSUM algorithm alarms.

The paper is structured as follows. First, the dynamic model and the estimation algorithm are briefly reviewed in Section 2. This is followed by a discussion on how to detect lane departures of leading vehicles and how this information can be used to obtain better estimates. In Section 4 it is discussed how the error caused by using the wrong model during the detection phase can be corrected. Finally, we provide a discussion on alternative methods in Section 5 and state our conclusion in Section 6.

2 Estimation Problem

The dynamic model is based on a curved, road-aligned coordinate system, defined in Figure 4, where x is the longitudinal position along the road and y is the lateral position perpendicular to x . For example, this means that if y^i is the lateral position of object i , then $y^i = 0$ simply means that object i is at the center of our own lane, irrespective of road shape. For the lateral dynamics, a constant position model is used, i.e., $\dot{y}^i = 0$, and for the longitudinal dynamics a constant velocity model is used. Other states in the model are lane width W , host vehicle lateral position y_{off} , host vehicle heading angle relative to the lane Ψ_{rel} , road curvature parameter c_0 , which is defined as the inverse road radius and finally the road clothoid parameter c_1 , i.e., the curvature change rate. The vision system delivers estimates of W , y_{off} , Ψ_{rel} and c_0 , which are used as measurements in our estimation problem. Furthermore, the radar provides measurements of the relative position of objects resolved in the coordinate system (\tilde{x}, \tilde{y}) , attached to the host vehicle. The dynamic model is discussed in more detail in the Appendix and the resulting estimation problem and its solution is treated in Eidehall (2004), Eidehall and Gustafsson (2004), Eidehall et al. (2005). Tuning of the process and measurement noise will not be discussed

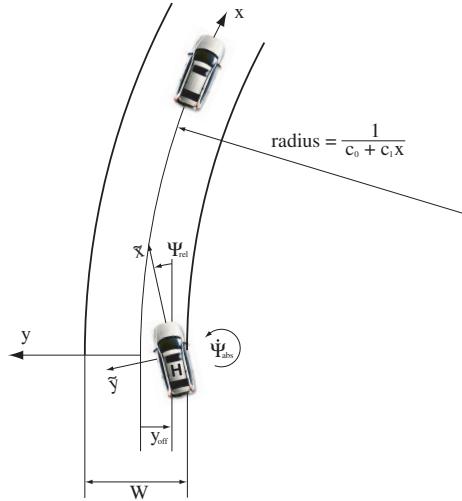


Figure 4: The surrounding vehicles are conveniently modeled and tracked using a curved, road-aligned coordinate system (x, y).

in detail, except for the process noise of y^i . The discrete-time dynamic model describing the evolution of y^i over time is given by

$$y_{t+1}^i = y_t^i + w_t^i, \quad (1)$$

where w_t^i is zero mean white Gaussian noise, with variance Q_{lat} . In applying an Extended Kalman Filter (EKF), the tuning parameter Q_{lat} describes to what degree it is believed that vehicles will keep driving at the same lateral position in relation to the lane.

3 Detecting Lane Departures

The approach employed for improving the road geometry estimates based on detecting lane departures is illustrated in Figure 5. This is a standard approach within the area of change detection, which is a well established research area, see, e.g., Gustafsson (2000), Basseville and Nikiforov (1993), Kay (1998). The aim of the detector in Figure 5 is to detect lane departures based on the information available in the residuals $\varepsilon_t = y_t - \hat{y}_t$ from the estimation algorithm. When a lane departure is detected this is indicated by an alarm from the detector, which is used to temporarily change model for the vehicle performing the lane departure. This implies that the estimation algorithm can provide a better estimate, simply due to the fact that a more accurate model is used. This section is concerned with devising the detection algorithm illustrated with the detection box in Figure 5. The estimation algorithm used in the present studies is based on the extended Kalman filter (Eidehall, 2004, Eidehall and Gustafsson, 2004). The basic components of a change detection algorithm are illustrated in Figure 6.

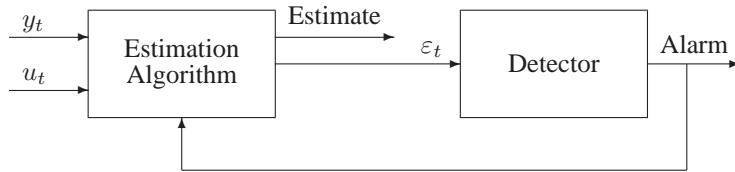


Figure 5: The estimation algorithm delivers residuals ε_t , which are used in the detector to decide whether a change has occurred or not. If a change is detected this information is fed back for use in the estimation algorithm. Note that in this application, one detector for each tracked vehicle is needed.

3.1 Distance Measure

The distance measure is used to assess whether a change has occurred or not. It is an important design variable, that should be chosen with the application in mind. Common standard choices are to use the residuals $s_t = \varepsilon_t$ or the squared residuals $s_t = \varepsilon_t^2$. However, in the present application this would provide poor detection performance. The reason is that the residuals only contain angular information. This would imply that the distance measure implicitly depend on the longitudinal distance to the leading vehicle, whereas for detecting lane departures we are only interested in lateral distances. If the longitudinal distance to the leading vehicle is small, a small change of its lateral position would lead to a large angular change. If the same change of lateral position would be observed for a vehicle further away, the angular change would be smaller. Hence, we need a distance measure that is invariant to the distance to the leading vehicle. The most natural choice in this respect is provided by lateral displacement of the leading vehicle, approximately given by

$$s_t = |\varepsilon_t r_t|, \quad (2)$$

where r_t denotes the distance to the leading vehicle, available from the estimation algorithm, primarily based on the radar measurements. The reason for using $|\varepsilon_t r_t|$ and not just $\varepsilon_t r_t$ in (2) is that we want to be able to detect both left and right lateral displacements, using a one-sided test.

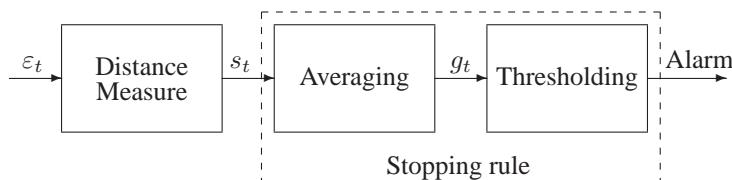


Figure 6: The components of the detector are a distance measure, and a stopping rule, where the latter consists of an averaging and a thresholding procedure.

3.2 Stopping Rule

The stopping rule is used to give an alarm when an auxiliary test statistic g_t exceeds a certain threshold. One of the most powerful tools for obtaining a good stopping rule in change detection problems is provided by the *CUSUM algorithm*, introduced by Page (1954).

Algorithm I.1 (CUSUM)

1. $g_t = g_{t-1} + s_t - \nu.$
2. If $g_t > h$: Alarm, $g_t = 0$ and $t_{\text{alarm}} = t$.
3. If $g_t < 0$: $g_t = 0$ and $\hat{t}_{\text{change}} = t$.

A rather detailed account of the CUSUM algorithm and its application in state estimation problems is provided in Gustafsson (2000). However, for the discussion to come we point out that the detection delay is the time delay between the actual event, in this case the start of a lane change manoeuvre, and the detection. In the CUSUM algorithm the detection delay is the time it takes for g_t to reach the threshold h , i.e., $t_{\text{alarm}} - \hat{t}_{\text{change}}$. This means that when an alarm is triggered, the actual event took place a certain time ago. We will get back to this fact in Section 4, where it is used to further enhance the estimation performance.

3.3 Application and Result

When the CUSUM algorithm gives an alarm this is fed back to the estimation algorithm, where an increased Q_{lat} is employed for the vehicle performing the lane departure. Since this model better describes the lane departure it will result in better estimates, which also is clear from Figure 7. This lane departure model is employed during an appropriate time, corresponding to a typical lane change. After this we switch back to the original model.

The idea outlined above has been tested using 35 minutes of authentic traffic data. The detection performance is detailed in Table 1. For the present application a missed

Table 1: Detection performance, based on 35 minutes of authentic traffic data.

Type	Number
Correct detections	35
Missed detections	3
False detections	27

detection is much worse than false detection. A missed detection clearly degrades the estimation performance substantially, see Figure 7, whereas a false detection merely implies a slight performance degradation, since more noise than necessary is used in the model. It is interesting, and perhaps not too surprising, to note that most of the false detections are due to sharp road turns. If these could be isolated, most of the false detections could probably be eliminated. However, since the false detections do not significantly degrade the performance this has not been investigated further.

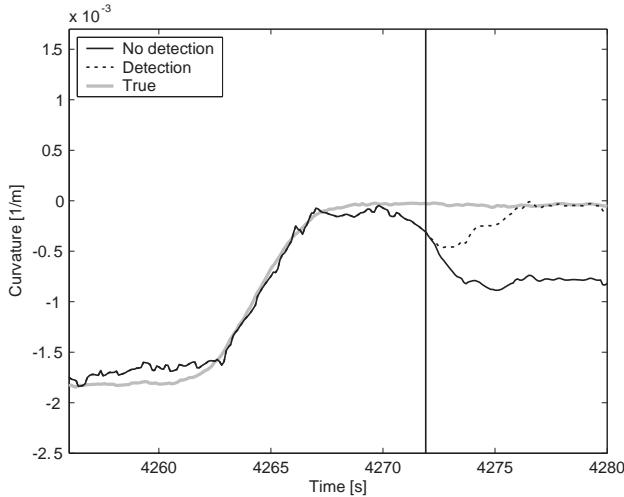


Figure 7: Illustrating how the estimation performance is improved using lane departure detection. This is the same data used in Figure 2, but the estimates from the filter based on change detection is also included.

4 Further enhancement

In this section, we introduce a way of correcting for the error that is caused due to the fact that the wrong model is used during the detection phase. The idea is to store old measurements y_t , input signals u_t , estimates $\hat{x}_{t|t}$ and covariance matrices $P_{t|t}$ in a memory. We propose a *refiltering* scheme, that on detection at time t_{alarm} , the filter is rerun with the correct model between times \hat{t}_{change} and t_{alarm} in order to correct for the error that is caused by using the wrong model. The estimate at time t_{alarm} is then replaced with the estimate that is obtained using the correct model. A schematic illustration of this idea is given in Figure 8.

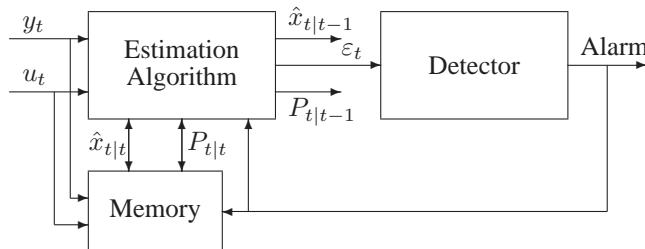


Figure 8: The change from Figure 5 is that a memory block has been included. The memory block stores the recent history of the measurements, input signals, estimates and their covariance.

In our application, this means that Q_{lat} is increased at time \hat{t}_{change} and then kept high according to the previous section so that the total time equals the time of a typical lane change. A result of this is typically a jump in the estimate at the detection times. Two detailed examples of the behavior of the enhanced algorithm are illustrated in Figure 9 and Figure 10. The performance for a five minute data set is shown in Figure 11. From

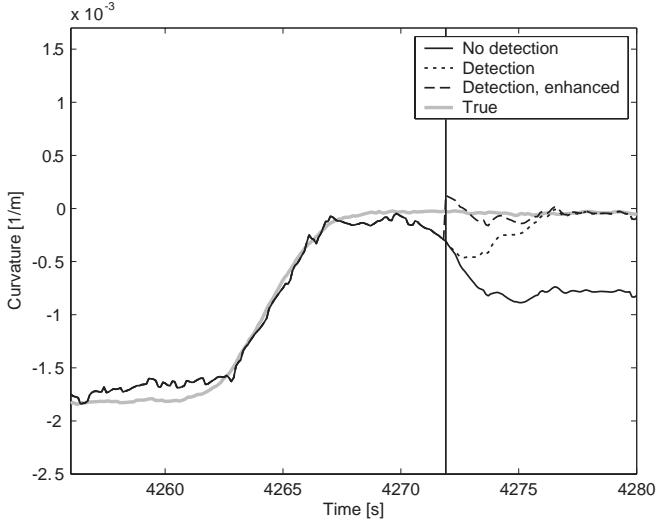


Figure 9: The behavior of the three approaches when the lane change is detected. The filter with no detection scheme deteriorates, the filter with detection converges when switching to the correct model, and the enhanced detection algorithm jumps to the value it would have had if it had used the correct model from the beginning of the lane change.

in this figure it is interesting to note that in the last turn, around time 4500 s, there is a time delay in the filter which is not present in any of the other turns. This is due to the fact that there are no vehicles to support the estimate and thus the curve can only be detected robustly once we have entered it ourselves.

5 Alternative methods

The paper by Weiss et al. (2004) discuss the use of a filter based on interacting multiple models (IMM) for detecting lane changes. The goal of their work is to improve the position estimates of surrounding vehicles, rather than road geometry. Of course, the same approach could be used in an integrated road geometry and object tracking model as the one proposed in this paper in order to also improve road geometry estimation.

In an IMM approach, two or more models are run simultaneously and they are each given a probability, of being the “correct model”, based on their residuals. The final estimate is then formed as a weighted average, using the probabilities as weights. We

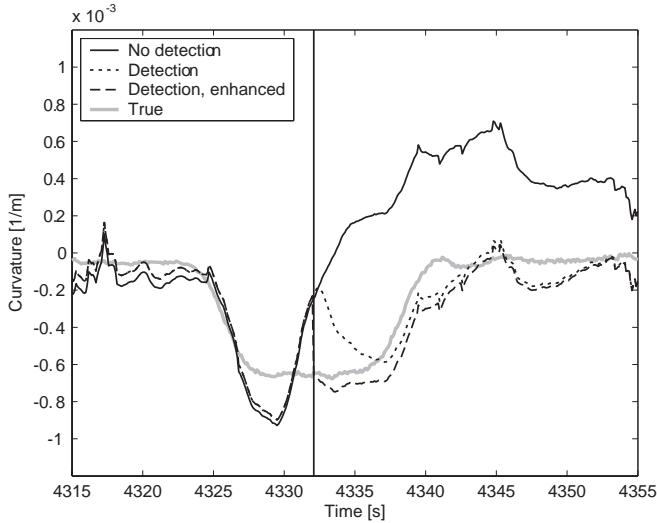


Figure 10: Same plots as in Figure 9 but for a different time interval.

believe that the methods we propose here, based on the CUSUM algorithm, have several advantages. Firstly, a lane change is a distinct event, so either one or the other model is

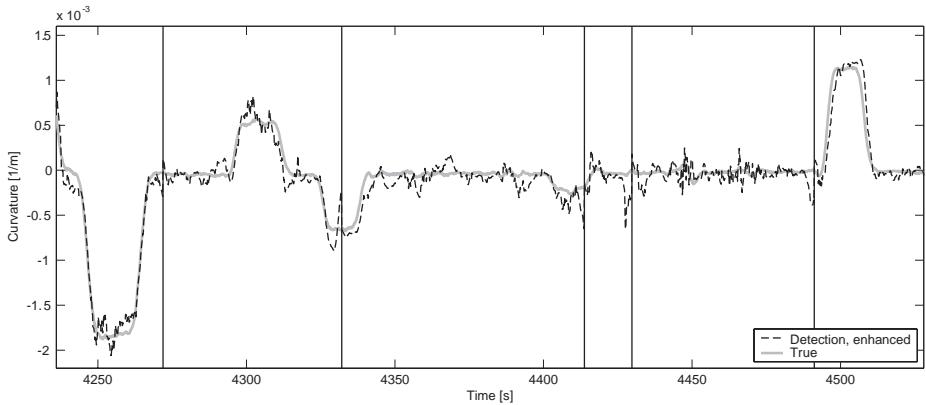


Figure 11: This figure shows the curvature estimate for a five minute data set collected in an authentic traffic environment, compared to the true curvature value. The vertical lines indicates detection of lane changes. It is interesting to note that in the last turn, around time 4500 s, there is a time delay in the filter which is not present in any of the other turns. This is due to the fact that there are no vehicles to support the estimate and thus the turn can only be detected robustly once we have entered it.

valid, not something in between. This means that conceptually, it is preferable to switch models completely rather than averaging two models. Secondly, the CUSUM algorithm provides a clear indication that something has happened, rather than a continuous change in probabilities and this indication can be used to take appropriate countermeasures. For example, this is necessary for initiating the refiltering scheme presented in Section 4.

Another idea that could be interesting to investigate is to use a two-sided test. In the proposed method, the absolute value of the residuals was used in combination with a one-sided test. An alternative could be to use the signed residuals and a two-sided test, which might eliminate some of the false alarms. The reason is that an alarm could be triggered by a driver who is "wobbling" in the lane but actually not changing lanes. On the other hand, it could be argued that we would benefit from detecting any kind of lateral movement, not just lateral movement related to a lane change.

6 Conclusion

By detecting behavior that deviates from the model in a tracking system, we can rely more on the model when it in fact is accurate. In the present application, this means that the road geometry estimate, which is supported by the motion of surrounding vehicles, can be significantly improved. A CUSUM algorithm is used, which has the advantage of giving a distinct alarm when a change has occurred. It is also concluded that the method of correcting for the error that was caused by using the wrong model during the detection phase does give further improvements of the estimation accuracy.

7 Acknowledgment

The authors would like to thank the SEnsor Fusion for Safety systems (SEFS) program, the Intelligent Vehicle Safety Systems (IVSS) program and Volvo Car Corporation for supporting this work.

Appendix – Dynamic Model

In this appendix the underlying dynamic model that is used throughout the paper is discussed in more detail. The derivation is performed in continuous-time. The discrete-time dynamics are obtained using the standard sampling formula (Rugh, 1996), under the assumption of piecewise constant input signals.

System Model

The coordinates x and y denote the position in the curved coordinate system, which is attached to the road according to Figure 4. The longitudinal coordinate x is relative, i.e., x is the longitudinal distance between the host vehicle and the tracked object. The motion model for the surrounding vehicles is greatly simplified in using the curved, rather than a Cartesian coordinate system. For example, it allows us to use the equation $\dot{y}^i = 0$, to model the assumption that the surrounding vehicles will follow their own lanes.

In the longitudinal direction $\dot{x}^i = -a \cos(\Psi_{\text{rel}})$ will be used, where a is the measured acceleration of the host vehicle, if available. If there are no measurements of the host vehicle's acceleration it is simply modeled as a random walk. Hence, we typically have the following motion model:

$$\dot{x}^i = v^i, \quad (3a)$$

$$\dot{v}^i = -a \cos(\Psi_{\text{rel}}), \quad (3b)$$

$$\dot{y}^i = 0, \quad (3c)$$

where v^i is the longitudinal relative velocity of vehicle i , i.e., the time derivative of x^i . It is affected by the host vehicle acceleration since it is the *relative* velocity that is modeled. For the road geometry parameters we first clarify that Ψ_{rel} is the angle between the host vehicle and the lane, see Figure 4, whereas Ψ_{abs} is the angle to some fix reference. A relationship between the two can be obtained by differentiating Ψ_{rel} w.r.t. time,

$$\Psi_{\text{rel}} = \Psi_{\text{abs}} + \gamma \Rightarrow \quad (4a)$$

$$\dot{\Psi}_{\text{rel}} = \dot{\Psi}_{\text{abs}} + \dot{\gamma} = \dot{\Psi}_{\text{abs}} + \frac{v}{r} = \dot{\Psi}_{\text{abs}} + c_0 v, \quad (4b)$$

where r is the current road radius, v the velocity and γ denotes the angle between the lane and some fix reference. $\dot{\Psi}_{\text{abs}}$ is typically measured using a yaw rate sensor. Furthermore,

$$\dot{y}_{\text{off}} = \sin(\Psi_{\text{rel}})v \approx \Psi_{\text{rel}}v. \quad (5)$$

Using $\dot{W} = 0$ and $\dot{c}_1 = 0$ continuous-time motion equations for the road model can be written

$$\dot{W} = 0, \quad (6a)$$

$$\dot{c}_0 = vc_1, \quad (6b)$$

$$\dot{c}_1 = 0, \quad (6c)$$

and for the motion of host vehicle we have

$$\dot{y}_{\text{off}} = v\Psi_{\text{rel}}, \quad (7a)$$

$$\dot{\Psi}_{\text{rel}} = vc_0 + \dot{\Psi}_{\text{abs}}. \quad (7b)$$

To account for uncertainties in the model we add zero mean white Gaussian noise to the corresponding discrete-time equations. The covariance matrices are Q_{road} , Q_{host} and Q_{obj} for the road, host and object states, respectively. Note that Q_{lat} , defined in Section 1 is the diagonal component of Q_{obj} corresponding to (3c), the lateral movement of the tracked vehicles.

Measurement Model

The measurements for the host vehicle are Ψ_{rel}^m , c_0^m , L^m and R^m , where the two latter are the distances to the left and right lane marking, see Figure 4. Superscript m is used

to denote measured quantities. The (relative) position $(\tilde{x}^m, \tilde{y}^m)$ of the surrounding vehicles is measured using radar. Note that the radar delivers measurements resolved in the Cartesian coordinate system, which is attached to the vehicle. The resulting measurement model is,

$$L^m = W/2 - y_{\text{off}}, \quad (8a)$$

$$R^m = -W/2 - y_{\text{off}}, \quad (8b)$$

$$\Psi_{\text{rel}}^m = \Psi_{\text{rel}}, \quad (8c)$$

$$c_0^m = c_0, \quad (8d)$$

$$\begin{pmatrix} \tilde{x}^{i,m} \\ \tilde{y}^{i,m} \end{pmatrix} = \frac{R(-\Psi_{\text{rel}})}{c_0} \begin{pmatrix} (1 + c_0 y^i) \sin(c_0 x^i) \\ (1 + c_0 y^i) \cos(c_0 x^i) - 1 - c_0 y_{\text{off}} \end{pmatrix}, \quad (8e)$$

where $R(-\Psi_{\text{rel}})$ is a rotational matrix performing clockwise rotation of Ψ_{rel} radians. Furthermore, zero mean white Gaussian measurement noise is added to (8). The covariance matrices are R_{host} and R_{obj} for the host/road and object states, respectively.

References

- Bar-Shalom, Y. and Li, X.-R. (1993). *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, Norwood, MA, USA.
- Basseville, M. and Nikiforov, I. V. (1993). *Detection of abrupt changes theory and applications*. Information and systems sciences series. Prentice Hall, Englewood Cliffs, NJ, USA.
- Dellaert, F. and Thorpe, C. (1997). Robust car tracking using Kalman filtering and Bayesian templates. In *Proceedings of the SPIE conference on Intelligent Transportation Systems*, volume 3207.
- Eidehall, A. (2004). *An Automotive Lane Guidance System*. Licentiate Thesis No 1122, Department of Electrical Engineering, Linköping University, Sweden.
- Eidehall, A. and Gustafsson, F. (2004). Combined road prediction and target tracking in collision avoidance. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 619–624, Parma, Italy.
- Eidehall, A. and Gustafsson, F. (2006). Obtaining reference road geometry parameters from recorded sensor data. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Tokyo, Japan. Submitted.
- Eidehall, A., Schön, T. B., and Gustafsson, F. (2005). The marginalized particle filter for automotive tracking applications. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 369–374, Las Vegas, USA.
- General (2000). Automotive collision avoidance system field operational test. Technical Report DOT HS 809 196, General Motors Corporation.

- Gustafsson, F. (2000). *Adaptive Filtering and Change Detection*. John Wiley & Sons, New York, USA.
- Kaempchen, N., Weiss, K., Schaefer, M., and Dietmayer, K. C. J. (2004). IMM object tracking for high dynamic driving maneuvers. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 825–830, Parma, Italy.
- Kay, S. M. (1998). *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*. Prentice Hall Signal Processing. Prentice Hall, Upper Saddle River, NJ, USA.
- Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1/2):100–115.
- Rugh, W. J. (1996). *Linear System Theory*. Information and system sciences series. Prentice Hall, Upper Saddle River, NJ, USA, second edition.
- Weiss, K., Kaempchen, N., and Kirchner, A. (2004). Multiple-model tracking for the detection of lane change maneuvers. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 937–942, Parma, Italy.
- Yen, P.-S., Fang, C.-Y., and Chen, S.-W. (2004). Motion analysis of nearby vehicles on a freeway. In *Proceedings of IEEE International Conference on Networking, Sensing & Control*, pages 903–908, Taipei, Taiwan.
- Zomotor, Z. and Franke, U. (1997). Sensor fusion for improved vision based lane recognition and object tracking with range-finders. In *Proceedings of IEEE Conference on Intelligent Transportation System*, pages 595–600, Boston, MA, USA.

Notation

Note that all vectors are column vectors. In general, lower case letters are used to denote vector valued and scalar variables and upper case letters are used for matrix valued variables. However, there might be exceptions from these general rules due to conventions. The same symbol can be used for different purposes. The principal notation is listed below, any deviations from this is explained in the text.

Symbols and Operators

x_t	State vector
y_t	Measurement signal
u_t	Known input signal
x_t^l	Linear state variable
x_t^n	Nonlinear state variable
x_t^p	State variable estimated using the particle filter
x_t^k	State variable estimated using the Kalman filter
e_t	Measurement noise
w_t	Process noise
ϑ	Parameter vector
θ	Parameter vector
P_t	Covariance matrix
Q_t	Covariance matrix for the process noise w_t
R_t	Covariance matrix for the measurement noise e_t
Y_N	Measurements up to time N , $\{y_1, y_2, \dots, y_N\}$
N	Length of the observed data set
\bar{x}_0	Initial value for the state x

\bar{P}_0	Initial value for the covariance P
\hat{x}	Estimator or estimate of x , determined by the context
$\hat{x}_{t t}$	Estimate of x at time t , given the information available at time t , i.e., the filtered estimate
$\hat{x}_{t+k t}$	Estimate of x at time $t + k$, given the information available at time t . $k < 0$ means smoothed estimate, and $k > 0$ means predicted estimate
ε_t	Residual vector, innovations
$f(\cdot)$	Equations for the system model
$h(\cdot)$	Equations for the measurement model
F_t	Linearized state update matrix
H_t	Linearized measurement relation matrix
K_t	Kalman gain
$\mathcal{N}(m, P)$	Normal (Gaussian) distribution with mean value m and covariance matrix P
$\mathcal{N}(x m, P)$	Normal (Gaussian) probability density function with mean value m and covariance matrix P
$\Phi(x)$	Normal (Gaussian) distribution function
$\mathcal{U}(a, b)$	Uniform distribution over the interval $[a, b]$
q	Unit quaternion
c_f	Position of the camera center, expressed in the F-system
v_t	Velocity of the camera center, expressed in the F-system
R_{cf}	Rotation matrix, expressing rotation from the F-system to the C-system
ω	Angular velocity
I_n	Unit matrix of dimension n
0_n	Null matrix of dimension n
M	Number of particles
$t(x)$	Target probability density function
$s(x)$	Sampling probability density function
$x_{t s}^{(i)}$	Particle i
$q_t^{(i)}$	Importance weight i
$\tilde{q}_t^{(i)}$	Normalized importance weight i
$\delta(\cdot)$	Dirac delta function
δ_{kl}	Kronecker delta function
$p_x(x)$	Probability density function of x
$p(x)$	Short form of above
$p_{x,y}(x, y)$	Joint probability density function of x and y
$p(x, y)$	Short form of above
$p_{x y}(x y)$	Conditional probability density function of x given y

$p(x y)$	Short form of above
$p_\theta(Y_N)$	Family of probability density functions, indexed by the parameter θ
$\hat{p}_M(\cdot)$	Probability density function approximated using M samples
L	Length of the sliding window
t	Time
T_s	Sample time
ν	Drift term in the CUSUM algorithm
$V_N(\theta, Y_N, U_N)$	Criterion function to be minimized
$L(\cdot)$	Log-likelihood function
$l(\cdot)$	Likelihood function
$g(\cdot)$	Inference function
$G(e^{i\omega})$	Transfer function
W	Lane width
Q_{lat}	Covariance matrix for the noise added to the lateral position of a vehicle
y_{off}	Host vehicle lateral position
Ψ_{rel}	Host vehicle heading angle relative to the lane
c_0	Road curvature parameter
c_1	Road clothoid parameter
\mathbf{R}^n	The set of real numbers in n dimensions
\mathbf{R}^+	The set of positive real numbers
$r_i[P]$	The i^{th} row degree of a polynomial matrix $P(s)$
$\text{diag}(a)$	A diagonal matrix with a as diagonal entry
\dot{x}	Time derivative of x
$\mathcal{R}(B)$	Range of the matrix B
\triangleq	Equal by definition
\sim	Denotes “is distributed according to”
\propto	Proportional to
\in	Belongs to
\forall	For all
$\xrightarrow{\text{a.s.}}$	Almost sure convergence
\xrightarrow{d}	Convergence in distribution
$\arg \min_x f(x)$	The value of x that minimizes $f(x)$
$\Pr_x(x \leq K)$	Probability that the random variable x is less than K
$\det A$	Determinant of matrix A
$\dim A$	Dimension of matrix A
$\text{Tr } A$	Trace of matrix A
A^T	Transpose of matrix A
A^{-1}	Inverse of matrix A

$\lfloor \cdot \rfloor$	Rounding downwards to nearest integer
$\text{sign}(x)$	Sign function
$\text{Cov}\{x\}$	Covariance matrix of the random variable x
$\text{E}\{x\}$	Expectation of the random variable x
$\text{E}\{x y\}$	Conditional expectation of the random variable x , given that the random variable $y = y$
$\text{E}_{\theta_k}\{x_t Y_N\}$	Expected value w.r.t. $p_{\theta=\theta'}(X_N Y_N)$
$\text{Var}\{x\}$	Variance of the random variable x
\min	Minimize
\max	Maximize
$\ x\ _A^2$	Weighted vector norm, $\ x\ _A^2 = x^T A x$
$ \cdot $	Absolute value

Abbreviations and Acronyms

a.s.	almost sure
flops	Floating-Point Operations
i.i.d.	independent identically distributed
pdf	probability density function
s.t.	subject to
w.r.t.	with respect to
ABS	Anti-lock Braking System
ACC	Adaptive Cruise Control
APP	Auxiliary Particle Filter
AR	Augmented Reality
ARMAX	AutoRegressive Moving Average with eXternal input
ARX	AutoRegressive with eXternal input
CUSUM	CUmulative SUM
DAE	Differential-Algebraic Equation
DCM	Direction Cosine Matrix
DGPS	Differential GPS
DME	Distance Measuring Equipment
EKF	Extended Kalman Filter
EF	Equivalent Flop
EM	Expectation Maximization
FIR	Finite Impulse Response
GIS	Geographic Information System
GPB	Generalized Pseudo-Bayesian
GPS	Global Positioning System
GS	Gaussian Sum
GPF	Gaussian Particle Filter
GSPF	Gaussian Sum Particle Filter
HMM	Hidden Markov Model

IMM	Interacting Multiple Model
IMU	Inertial Measurement Unit
INS	Inertial Navigation Systems
IR	Infrared
IS	Importance Sampling
KF	Kalman Filter
LMI	Linear Matrix Inequality
LS	Least-Squares
MAP	Maximum A Posteriori
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
MEMS	MicroElectronic Mechanical Systems
MFD	Matrix Fraction Description
MHE	Moving Horizon Estimation
ML	Maximum Likelihood
MMS	Minimum Mean Square
MMSE	Minimum Mean Square Error
MPC	Model Predictive Control
MPF	Marginalized Particle Filter
MSE	Mean Square Error
MV	Minimum Variance
NARMAX	Nonlinear ARMAX
ODE	Ordinary Differential Equation
PF	Particle Filter
POI	Point-of-interest
QP	Quadratic Program
RLS	Recursive Least Squares
RMSE	Root Mean Square Error
RPEKF	Range Parameterized Extended Kalman Filter
RPF	Regularized Particle Filter
RTS	Rauch-Tung-Striebel
SDAE	Stochastic Differential-Algebraic Equation
SIR	Sampling Importance Resampling
SIS	Sequential Importance Sampling
SLAM	Simultaneous Localization and Mapping
SO	Special Orthogonal
SVD	Singular Value Decomposition
TAP	Terrain Aided Positioning
UKF	Unscented Kalman Filter
WLS	Windowed Least Squares
YALMIP	Yet Another LMI Parser

Index

- acceleration, 8, 201, 203, 209, 252
accelerometer, 8, 202–204
acceptance – rejection sampling, 55–57, 190
acceptance probability, 53, 190
active safety system, 2
adaptive cruise control, 2, 230
aircraft navigation, see navigation system
aircraft positioning, see positioning applications
alarm, 44, 245, 247, 251
angular velocity, 8, 201, 204, 207
anti-lock braking system, 2
artificial marker, 7, 201, 202, 208
ARX model, 162
augmented reality, 201
automotive positioning, see positioning applications
automotive safety system, 2
automotive sensors, 3
automotive target tracking, see target tracking applications
auxiliary test statistic, see change detection
auxiliary particle filter, see particle filter
azimuth, 136, 231, 232
band limited noise, 146
Bayes’ theorem, 33, 111, 132, 163, 174, 175, 190, 191
Bayesian bootstrap, 173
bearing, 226
bearings-only target tracking, see target tracking applications
bias estimation, 227
bilinear model, 178, 186
black box system identification, see system identification
bootstrap, 54
burn-in, 56
camera, 7, 201, 206
camera center, 203, 208
model, 206
pinhole model, 206
Cartesian coordinate system, 226, 230
central limit theorem, 52, 110, 174
change detection, 43, 49, 245
auxiliary test statistic, 44, 247
averaging, 44, 246
distance measure, 44, 246
stopping rule, 44, 246, 247
thresholding, 44, 246
chaotic model, 179
Chapman–Kolmogorov equation, 34
collision avoidance, 2, 3, 107, 116, 230, 241
colored noise, 115

- computer controlled system, 146
 concave function, 160
 conditionally linear, 107, 111
 constrained state estimation, see state estimation
 constraints, 45, 47, 160
 continuous-time white noise, 147
 convex optimization, 45, 47, 159, 160
 convex optimization estimation, 161
 convex optimization filtering, 48, 162, 163
 coordinate system, 203
 coordinated turn model, 4, 227
 correlated noise, 123
 Cramér-Rao, 106, 224
 cross-covariance, 114, 150
 cumulative sum (CUSUM), 44, 45, 243
 curve fitting, 45
 curved coordinate system, 5, 230, 242, 244, 251
 damped Gauss – Newton, 186
 dead-reckoning, 201, 202
 depth, 207
 depth database, 227
 descriptor system, 22, 146, 147, 153, 162
 detection, 6, 247
 diagonal model, 109
 differential GPS, 121, 227
 differential-algebraic equation, 22, 145, 152
 example, 151
 linear, 29, 145
 standard form, 30, 147, 152
 stochastic, 23, 29, 145
 Dirac delta function, 51
 Direction Cosine Matrix, 204
 distance measure, see change detection
 distance measurement equipment, 227
 disturbance, 146
 drift term, 45
 Dymola, see object-oriented modeling
 effective sample size, 60
 empirical density function, 51, 52
 equivalent flop, 12, 77, 131, 134, 221
 Euler angles, 205
 Euler approximation, 208
 Euler symmetric parameters, 205
 Euler-Rodrigues parameters, 205
 expectation maximization, 72, 185–188
 system identification, 75
 experiment design, 70
 extended Kalman filter, see Kalman filter
 external signal, 22
 fault diagnosis, 49
 feature, 8, 202, 206
 feature displacement, 202, 206, 208
 feature extraction, 9, 206
 feedback, 2
 fictitious measurement, 165
 filter banks, 49
 fixed-interval smoothing, see smoothing
 fixed-lag smoothing, see smoothing
 fixed-point smoothing, see smoothing
 floating-point operations, 133
 focal length, 206
 forward-backward smoothing, 67
 free-D system, 209
 Gauss – Newton, 195
 Gaussian particle filter, see particle filter
 Gaussian sum, 39, 217
 Gaussian sum particle filter, see particle filter
 generalized causality principle, 24
 generalized pseudo-Bayesian, 39
 generalized real Schur form, 148
 generalized Sylvester equation, 148
 generalized system, 22, 146
 geographic information system, 116, 226, 227
 global maximum, 195
 global optimum, 47, 160
 Gram–Schmidt procedure, 123, 132
 gravity, 203
 gray box identification, see system identification
 gyroscope, 8, 202, 203, 207
 Hammerstein – Wiener structure, 185
 heading, 119
 hidden Markov model, 25, 159, 164
 Hilbert space, 40

- homography, 202
image feature, 206, 209
image plane, 206, 208
image processing, 3, 201
implicit measurement, 23, 207
implicit system, 22, 146
importance function, 53, 60
importance sampling, 53, 57, 190
importance weight, 54, 58
impulse controllability, 149
impulse response, 21
incomplete data, 72, 187
inertial forces, 203
inertial measurement unit, 8, 202
inertial navigation system, 106, 118, 226, 228
inertial sensor, 201, 203, 206, 208
inertial system, 203
inference function, 174
infrared, 226, 231
initial parameter, 193
innovation form, 162, 178
innovation process, 40, 43
instrument landing system, 202
integrated aircraft navigation, 118
integrated navigation, 201
interacting multiple model, 39, 243, 249
internal variable, 22, 146, 150
Itô calculus, 23
iterated EKF, 38

Jacobian, 186
Jensen's inequality, 73, 188
jittering, 60
jump Markov linear system, 113

Kalman filter, 11, 41, 63, 64, 150, 174, 186
 descriptor system, 154
 differential-algebraic equation, 151
 extended, 4, 5, 11, 37, 38, 75, 106, 117, 217, 245
 linearized, 37
 unscented, 39
Kronecker canonical form, 30
Kronecker delta function, 29
Kullback-Leibler information, 73

lane guidance, 3, 230
Laplace transform, 147
latitude, 229
law of large numbers, 52
least-squares, 32, 46
 weighted, 41, 45, 163
 windowed, 48
likelihood function, 57, 186, 224
limit distribution, 53
linear regression, 75, 161, 177, 178
linear state-space model, see state-space model
linear sub-structure, 27, 51, 63, 107, 115, 122, 131, 215, 217
linearized Kalman filter, see Kalman filter
local minima, 75, 166, 186, 195
local optimum, 47, 160
log-concave, 48, 159, 160, 163, 164
log-likelihood, 71, 73, 188
longitude, 229

manifold, 205
map information, 23
marginalization, 64, 72, 107, 108, 162, 172, 174
marginalize, 161
marginalized particle filter, 4, 5, 28, 63, 64, 72, 75, 105, 108, 131, 171, 174, 215, 217, 219
 complexity analysis, 131, 133, 221
 quantization, 223
markerbased, 201
markerless, 202
Markov chain Monte Carlo method, 51, 53, 65
Markov property, 24, 33, 111, 190
matrix fraction description, 149
maximum a posteriori, 36, 46, 159, 160
maximum likelihood, 10, 36, 71, 185–187, 224
measure theory, 25
measurement model, 10, 22, 24, 252
measurement noise, 10, 25
measurement update, 41
Metropolis – Hastings algorithm, 56

- Metropolis – Hastings independence sampling, 56, 190
- midriser quantizer, 223
- minimum mean square error, 36
- minimum variance, 36
- missing data, 47
- missing data, 72, 75, 187, 188
- mixed linear/nonlinear state-space model, see state-space model
- model, 21
- model predictive control, 45, 47, 48
- model uncertainty, 47
- Modelica, see object-oriented modeling
- motion model, 5, 202, 226, 251
- moving horizon estimation, 48
- NARMAX, 185
- navigation system, 202
- aircraft, 105, 107, 202
 - augmented reality, 7, 201
 - automotive, 2, 3
 - underwater, 107
- neural network, 185
- Newton's second law, 204
- nilpotent matrix, 30, 147
- non-convex, 47, 49, 72, 160, 187, 205
- noncausal system, 146
- nonlinear smoothing, 189
- nonlinear state estimation, see
 state estimation
- nonlinear state-space model, see
 state-space model
- nonlinear system identification, see
 system identification
- nuisance variable, 161, 208
- object-oriented modeling, 22, 145
- Dymola, 22, 145
 - Modelica, 22, 145
 - Omola, 22, 145
 - SimMechanics, 145
- observer, 42
- observer canonical form, 178
- Omola, see object-oriented modeling
- optical distortion, 206
- ordinary differential equation, 23, 146, 148
- orientation, 7, 204, 205
- parameter estimation, see system identification
- particle smoother, 192
- particle filter, 11, 40, 56, 58, 64, 105, 108, 171, 173, 174, 186, 189, 191, 207, 218
- algorithm modifications, 60
 - auxiliary, 60, 115
 - Gaussian, 60, 115
 - Gaussian sum, 60
 - implementation, 61
 - regularized, 60
 - resample, 58, 62
 - system identification, 74, 75, 171
- particle methods, see sequential Monte Carlo methods
- particle smoother, 65, 66, 75, 185, 191
- passive safety system, 2
- perfect sampling, 52
- pinhole model, 206
- pitch, 119
- point estimate, 35, 68
- point-mass filter, 39, 106, 217
- polynomial matrix, 149
- row degree, 149
 - row reduced, 149
- pose, 208
- positioning, 23, 107
- positioning applications, 116, 225–227
- aircraft, 226, 228
 - automotive, 226, 230
 - underwater, 226, 227
- prediction error method, 71
- prior, 24, 36
- prior information, 159, 164
- process noise, 10, 25
- proper, 150
- quadratic program, 46, 47
- quantization, 223
- radar, 4, 136, 223, 226, 230, 232, 241, 244, 253
- radar altimeter, 119, 228

- radar target tracking, see target tracking applications
random number generation, 53, 189
random walk, 75, 171, 172
range, 136
range parameterized extended Kalman filter, 39, 231
Rao-Blackwellization, 64, 107, 174
Rao-Blackwellized particle filter, see marginalized particle filter
Rauch-Tung-Striebel formulas, 43
refiltering, 248
regularized particle filter, see particle filter
resample, 54, 58
residual, 245
residual sampling, 59
Riccati recursion, 117, 135, 165, 176, 222
rigid body, 204
road geometry, 3, 4, 6, 226, 230, 241
road model, 252
robust estimation, 48
roll, 119
root mean square error, 139
rotating masses, 151
rotation parameterization, 205
rotation group, 205
rotation matrix, 204
roughening noise, 60, 171, 172, 175, 179
- sample impoverishment, 59
sampling density, 53, 190
sampling importance resampling, 53, 55
scene model, 9, 202
see-through head-mounted display, 7
semi-state system, 22, 146
sensor fusion, 1, 3, 6, 9, 10, 225, 234, 242
sensor network, 223
sensor offset, 115
sequential Monte Carlo method, 11, 39, 51, 74
SimMechanics, see object-oriented modeling
simple random resampling, 59
singular system, 22
singular value decomposition, 148
smoothing, 42, 65, 188
fixed-interval smoothing, 42
fixed-lag smoothing, 43
fixed-point smoothing, 43
sonar, 226, 227
spline, 39
square-root factorization, 42, 43
state augmentation, 75, 172
state estimation, 31, 159
change detection, 43, 245
constrained, 47, 159, 160
convex optimization, 45, 159
deterministic, 45
linear, 40
nonlinear, 8, 10, 31, 36, 75
smoothing, 42, 65, 164, 186, 189
state-space model, 23, 25, 175
linear, 29, 164
mixed linear/nonlinear, 4, 27, 64, 105
nonlinear, 10, 25, 105
steering wheel angle, 5
stochastic integration, 51
stopping rule, see change detection
stratified sampling, 59
strictly white noise, 26
subspace identification, see system identification
system identification, 24, 69, 162, 188
bilinear, 74
black box, 70
differential-algebraic equation, 71
expectation maximization, 72, 74, 75, 185, 189
experiment design, 70
gray box, 70, 178
marginalized particle filter, 75, 171
nonlinear, 11, 69, 171, 185, 186
particle filter, 171
particle methods, 74
subspace, 178
subspace method, 71, 171
validation, 70
system model, 10, 24, 251
systematic sampling, 59, 60
- target density, 53, 189

target tracking applications, 107, 116, 136,
225, 226, 230
automotive, 226, 230
bearings-only, 117, 226, 231
radar, 136, 226, 232
Taylor expansion, 38, 227
terrain elevation database, 119, 210, 228,
229
terrain navigation, 116
terrain-aided positioning, 106, 118
time update, 41
transfer function, 21, 149, 162
 causal, 155
 proper, 149, 150
transition probability, 164
triangular model, 111, 117
two-filter formula, 34, 67

underwater depth map, 227
underwater positioning, see positioning applications
unimodular matrix, 149
unit quaternion, 203, 205
unmodeled dynamics, 146
unscented Kalman filter, see Kalman filter

validation, 70
variance reduction, 174, 221
vector space, 205
vision, 4, 5, 8, 206, 223, 226, 230, 241, 244
Volterra series, 21, 185

weight matrix, 46
weighted bootstrap, 54
weighted least-squares, see least-squares
wheel speed, 5, 226, 230
white noise, 26, 146, 148, 160
whiteness test, 49
Wiener – Kolmogorov filtering theory, 32
windowed least-squares, see least-squares

yaw rate, 5
yaw rate sensor, 252

PhD Dissertations
Division of Automatic Control
Linköpings universitet

- M. Millnert:** Identification and control of systems subject to abrupt changes. Thesis No. 82, 1982. ISBN 91-7372-542-0.
- A. J. M. van Overbeek:** On-line structure selection for the identification of multivariable systems. Thesis No. 86, 1982. ISBN 91-7372-586-2.
- B. Bengtsson:** On some control problems for queues. Thesis No. 87, 1982. ISBN 91-7372-593-5.
- S. Ljung:** Fast algorithms for integral equations and least squares identification problems. Thesis No. 93, 1983. ISBN 91-7372-641-9.
- H. Jonson:** A Newton method for solving non-linear optimal control problems with general constraints. Thesis No. 104, 1983. ISBN 91-7372-718-0.
- E. Trulsson:** Adaptive control based on explicit criterion minimization. Thesis No. 106, 1983. ISBN 91-7372-728-8.
- K. Nordström:** Uncertainty, robustness and sensitivity reduction in the design of single input control systems. Thesis No. 162, 1987. ISBN 91-7870-170-8.
- B. Wahlberg:** On the identification and approximation of linear systems. Thesis No. 163, 1987. ISBN 91-7870-175-9.
- S. Gunnarsson:** Frequency domain aspects of modeling and control in adaptive systems. Thesis No. 194, 1988. ISBN 91-7870-380-8.
- A. Isaksson:** On system identification in one and two dimensions with signal processing applications. Thesis No. 196, 1988. ISBN 91-7870-383-2.
- M. Viberg:** Subspace fitting concepts in sensor array processing. Thesis No. 217, 1989. ISBN 91-7870-529-0.
- K. Forsman:** Constructive commutative algebra in nonlinear control theory. Thesis No. 261, 1991. ISBN 91-7870-827-3.
- F. Gustafsson:** Estimation of discrete parameters in linear systems. Thesis No. 271, 1992. ISBN 91-7870-876-1.
- P. Nagy:** Tools for knowledge-based signal processing with applications to system identification. Thesis No. 280, 1992. ISBN 91-7870-962-8.
- T. Svensson:** Mathematical tools and software for analysis and design of nonlinear control systems. Thesis No. 285, 1992. ISBN 91-7870-989-X.
- S. Andersson:** On dimension reduction in sensor array signal processing. Thesis No. 290, 1992. ISBN 91-7871-015-4.
- H. Hjalmarsson:** Aspects on incomplete modeling in system identification. Thesis No. 298, 1993. ISBN 91-7871-070-7.
- I. Klein:** Automatic synthesis of sequential control schemes. Thesis No. 305, 1993. ISBN 91-7871-090-1.
- J.-E. Strömberg:** A mode switching modelling philosophy. Thesis No. 353, 1994. ISBN 91-7871-430-3.
- K. Wang Chen:** Transformation and symbolic calculations in filtering and control. Thesis No. 361, 1994. ISBN 91-7871-467-2.
- T. McKelvey:** Identification of state-space models from time and frequency data. Thesis No. 380, 1995. ISBN 91-7871-531-8.
- J. Sjöberg:** Non-linear system identification with neural networks. Thesis No. 381, 1995. ISBN 91-7871-534-2.
- R. Germundsson:** Symbolic systems – theory, computation and applications. Thesis No. 389, 1995. ISBN 91-7871-578-4.
- P. Pucar:** Modeling and segmentation using multiple models. Thesis No. 405, 1995. ISBN 91-7871-627-6.
- H. Fortell:** Algebraic approaches to normal forms and zero dynamics. Thesis No. 407, 1995. ISBN 91-7871-629-2.

- A. Helmersson:** Methods for robust gain scheduling. Thesis No. 406, 1995. ISBN 91-7871-628-4.
- P. Lindskog:** Methods, algorithms and tools for system identification based on prior knowledge. Thesis No. 436, 1996. ISBN 91-7871-424-8.
- J. Gunnarsson:** Symbolic methods and tools for discrete event dynamic systems. Thesis No. 477, 1997. ISBN 91-7871-917-8.
- M. Jirstrand:** Constructive methods for inequality constraints in control. Thesis No. 527, 1998. ISBN 91-7219-187-2.
- U. Forssell:** Closed-loop identification: Methods, theory, and applications. Thesis No. 566, 1999. ISBN 91-7219-432-4.
- A. Stenman:** Model on demand: Algorithms, analysis and applications. Thesis No. 571, 1999. ISBN 91-7219-450-2.
- N. Bergman:** Recursive Bayesian estimation: Navigation and tracking applications. Thesis No. 579, 1999. ISBN 91-7219-473-1.
- K. Edström:** Switched bond graphs: Simulation and analysis. Thesis No. 586, 1999. ISBN 91-7219-493-6.
- M. Larsson:** Behavioral and structural model based approaches to discrete diagnosis. Thesis No. 608, 1999. ISBN 91-7219-615-5.
- F. Gunnarsson:** Power control in cellular radio systems: Analysis, design and estimation. Thesis No. 623, 2000. ISBN 91-7219-689-0.
- V. Einarsson:** Model checking methods for mode switching systems. Thesis No. 652, 2000. ISBN 91-7219-836-2.
- M. Norrlöf:** Iterative learning control: Analysis, design, and experiments. Thesis No. 653, 2000. ISBN 91-7219-837-0.
- F. Tjärnström:** Variance expressions and model reduction in system identification. Thesis No. 730, 2002. ISBN 91-7373-253-2.
- J. Löfberg:** Minimax approaches to robust model predictive control. Thesis No. 812, 2003. ISBN 91-7373-622-8.
- J. Roll:** Local and piecewise affine approaches to system identification. Thesis No. 802, 2003. ISBN 91-7373-608-2.
- J. Elbornsson:** Analysis, estimation and compensation of mismatch effects in A/D converters. Thesis No. 811, 2003. ISBN 91-7373-621-X.
- O. Härkegård:** Backstepping and control allocation with applications to flight control. Thesis No. 820, 2003. ISBN 91-7373-647-3.
- R. Wallin:** Optimization algorithms for system analysis and identification. Thesis No. 919, 2004. ISBN 91-85297-19-4.
- D. Lindgren:** Projection methods for classification and identification. Thesis No. 915, 2005. ISBN 91-85297-06-2.
- R. Karlsson:** Particle Filtering for Positioning and Tracking Applications. Thesis No. 924, 2005. ISBN 91-85297-34-8.
- J. Jansson:** Collision Avoidance Theory with Applications to Automotive Collision Mitigation. Thesis No. 950, 2005. ISBN 91-85299-45-6.
- E. Geijer Lundin:** Uplink Load in CDMA Cellular Radio Systems. Thesis No. 977, 2005. ISBN 91-85457-49-3.
- M. Enqvist:** Linear Models of Nonlinear Systems. Thesis No. 985, 2005. ISBN 91-85457-64-7.