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

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 komplicerade 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 kombination 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 målföljningstillämpningar. 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 överlagra datorgenererade 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

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 state[1] 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

[1] Depending 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.

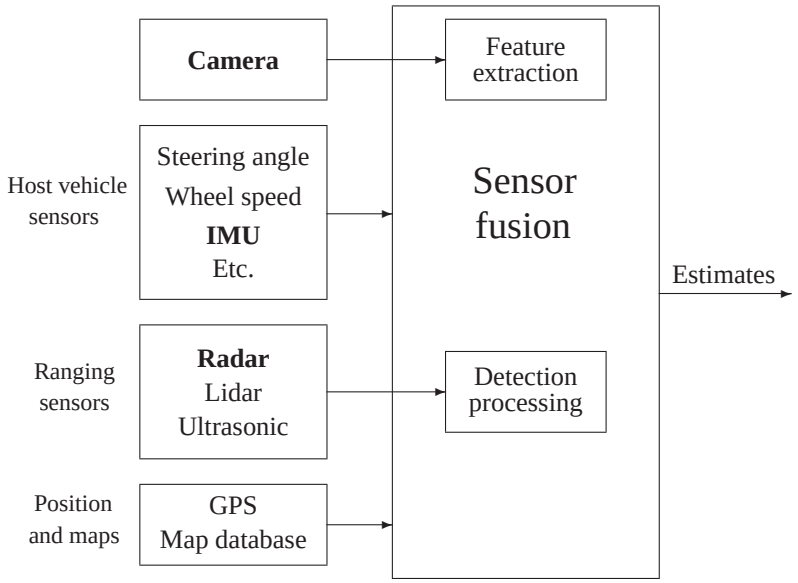

Figure 1.1: *The most important factor enabling future automotive safety systems is the availability of accurate information about the state. The process of obtaining this information is to a large extent dependent on a unified treatment of the sensor information, as illustrated in this figure. The aim 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, SEFS[2], 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), Nwagboso (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

[2] *SEnsor 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 $y^i_{t+1} = y^i_t + w_t, w_t \sim \mathcal{N}(0, Q_{\text{lat}})$. 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

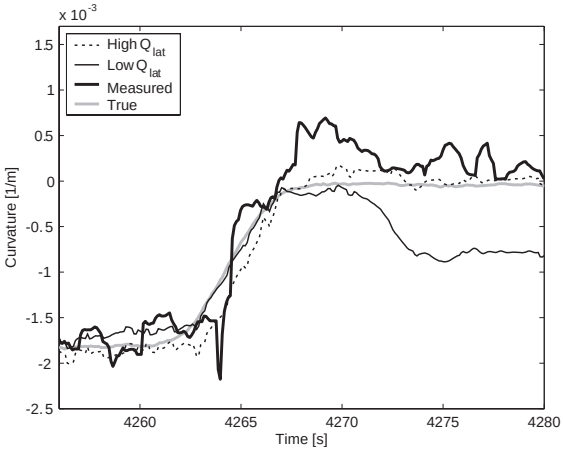

Figure 1.3: *Comparison of estimation performance from two filters, one with a large Q_{lat} and one with a small Q_{lat}. The raw measurement signal from the image processing unit is also included. Comparing this raw vision measurement to the result from the filters clearly illustrates the power of 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 $4270\ \text{s}$, 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* $4270\ \text{s}$*, 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).

(a) *Visualization of virtual objects in a live broadcast. Courtesy of BBC R&D.*

(b) *Assistance during maintenance. Courtesy of Fraunhofer IGD.*

(c) *Adding virtual graphics to sports scenes. Courtesy of BBC R&D.*

(d) *Visualization of virtual reconstructions of archaeological sites. Courtesy of Fraunhofer IGD.*

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

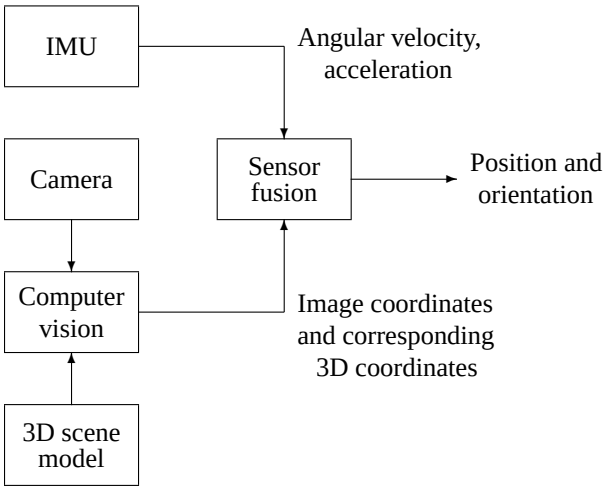

Figure 1.6: *Schematic illustration of the approach. The sensor fusion module is basically a recursive nonlinear state estimator, using information from the inertial measurement unit (IMU) and the computer vision system to compute an estimate of the position and orientation of the camera.*

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,

[3]Markerless 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

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

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

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:

> 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*,
>
> 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

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

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)

$$H(y(t_k), z(t_k), u(t_k), e(t_k), \theta, t_k) = 0, \tag{2.2}$$

[1] Other 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

$$y(t_k) = h(z(t_k), u(t_k), e(t_k), \theta, t_k), \tag{2.3}$$

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

$$F(\dot{z}(t), z(t), u(t), w(t), \theta, t) = 0, \tag{2.4a}$$
$$H(y(t_k), z(t_k), u(t), e(t_k), \theta, t_k) = 0, \tag{2.4b}$$

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,

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

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 is 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*

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

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

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

where we have made use of the Markov property. The notation $p_\theta(x)$ is used 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*,

$$y_t \sim p_\theta(y_t|x_t). \tag{2.8}$$

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

$$p_\theta(y_t|x_1, \ldots, x_N) = p_\theta(y_t|x_t), \qquad \forall t,\ 1 \leq t \leq N. \tag{2.9}$$

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

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

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

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

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

$$x_{t+1} = f(x_t, w_t, \theta, t), \tag{2.12a}$$
$$y_t = h(x_t, e_t, \theta, t), \tag{2.12b}$$

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

$$x_{t+1} = f(x_t, \theta, t) + w_t, \tag{2.13a}$$
$$y_t = h(x_t, \theta, t) + e_t, \tag{2.13b}$$

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,

$$p_\theta\left(x_{t+1}|x_t\right) = p_{w_t}(x_{t+1} - f(x_t, \theta, t)), \tag{2.14a}$$
$$p_\theta\left(y_t|x_t\right) = p_{e_t}(y_t - h(x_t, \theta, t)). \tag{2.14b}$$

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,

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

Furthermore, using conditioning and the Markov property we have

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

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,

$$\mathrm{E}\left\{(w_t - \mathrm{E}\left\{w_t\right\})(w_s - \mathrm{E}\left\{w_s\right\})^T\right\} = 0, \qquad t \neq s. \tag{2.17}$$

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*

$$p(w_t, w_s) = p(w_t)p(w_s), \qquad t \neq s. \tag{2.18}$$

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

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

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

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

$$x_t = \begin{pmatrix} x_t^l \\ x_t^n \end{pmatrix}, \tag{2.20}$$

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

$$x_{t+1}^n = f^n(x_t^n, t) + A^n(x_t^n, t)x_t^l + G^n(x_t^n, t)w_t^n, \tag{2.21a}$$
$$x_{t+1}^l = f^l(x_t^n, t) + A^l(x_t^n, t)x_t^l + G^l(x_t^n, t)w_t^l, \tag{2.21b}$$
$$y_t = h(x_t^n, t) + C(x_t^n, t)x_t^l + e_t, \tag{2.21c}$$

where the process noise is assumed white and Gaussian distributed with

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

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 linear[2]. 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.

$$x_{t+1}^n = A_{n,t}^n x_t^n + A_{l,t}^n x_t^l + G_t^n w_t^n, \tag{2.23a}$$
$$x_{t+1}^l = A_{n,t}^l x_t^n + A_{l,t}^l x_t^l + G_t^l w_t^l, \tag{2.23b}$$
$$y_t = h(x_t^n, t) + e_t, \tag{2.23c}$$

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).

[2] Strictly 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

$$x_{t+1} = A_t(\theta)x_t + w_t, \tag{2.24a}$$
$$y_t = C_t(\theta)x_t + e_t, \tag{2.24b}$$

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

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

$$E(\theta)\dot{z}(t) + F(\theta)z(t) = B_w(\theta)w(t), \tag{2.25a}$$
$$y(t_k) = C(\theta)z(t_k) + e(t_k), \tag{2.25b}$$

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

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

allows us to write (2.25) *as*

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

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). □

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

$$\dot{x}_1(t) = Ax_1(t) + G_1 w(t), \tag{2.28a}$$

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

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

$$NG_2 = 0. \tag{2.29}$$

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

$$I(g(x_t)) \triangleq \mathrm{E}_{p(x_t|Y_s)}\left\{g(x_t)\right\} = \int_{\mathbf{R}^{n_x}} g(x_t)p(x_t|Y_s)\,dx_t. \tag{3.1}$$

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 Gieseking (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 determination 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 where given on how to

[1] In 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

$$x_{t+1} \sim p(x_{t+1}|x_t), \tag{3.2a}$$
$$y_t \sim p(y_t|x_t). \tag{3.2b}$$

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

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

Consider the filtering density,

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

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

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

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,

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

resulting in the following expression

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

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 $p(x_{t+k}, \ldots, x_t|Y_t)$ w.r.t. $X_{t:t+k-1} = \{x_i\}_{i=t}^{t+k-1}$. Hence,

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

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

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

where

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

Furthermore,

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

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

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

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*

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

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

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

where

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

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.

$$m_t : U_s \times Y_s \to \mathbf{R}^{n_x} \tag{3.14}$$

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 used 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

$$\hat{x}^{MV} \triangleq \arg\min_{\hat{x}} \ \mathrm{E}\left\{\|x-\hat{x}\|^2 \,|y\right\} \tag{3.15}$$

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

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

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

$$\hat{x}^{MV} = \mathrm{E}\left\{x|y\right\} = \int x p(x|y)\,dx. \tag{3.17}$$

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,

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

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,

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

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

$$x_{t+1} = f(x_t, t) + w_t, \qquad w_t \sim \mathcal{N}(0, Q_t), \tag{3.20a}$$
$$y_t = h(x_t, t) + e_t, \qquad e_t \sim \mathcal{N}(0, R_t). \tag{3.20b}$$

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,

$$f(x_t, t) \approx f(\hat{x}_{t|t}, t) + \left.\frac{\partial f(x,t)}{\partial x}\right|_{x=\hat{x}_{t|t}} (x_t - \hat{x}_{t|t}), \tag{3.21a}$$

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

Using this approximation in (3.20) gives

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

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

where

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

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*

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

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

where

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

$$P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}, \tag{3.25b}$$

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

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

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

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

$$F_t = \left.\frac{\partial f(x,t)}{\partial x}\right|_{x_t=\hat{x}_{t|t}}, \qquad H_t = \left.\frac{\partial h(x,t)}{\partial x}\right|_{x_t=\hat{x}_{t|t-1}}. \tag{3.26}$$

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

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

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,

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

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,

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

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

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, $\hat{x}_{t|s} = \mathrm{E}\{x_t | Y_s\}$.

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*

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

where

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - C_t\hat{x}_{t|t-1} - D_t u_t), \tag{3.31a}$$
$$P_{t|t} = P_{t|t-1} - K_t C_t P_{t|t-1}, \tag{3.31b}$$
$$\hat{x}_{t+1|t} = A_t\hat{x}_{t|t} + B_t u_t, \tag{3.31c}$$
$$P_{t+1|t} = A_t P_{t|t} A_t^T + Q_t, \tag{3.31d}$$
$$K_t = P_{t|t-1}C_t^T(C_t P_{t|t-1} C_t^T + R_t)^{-1}, \tag{3.31e}$$

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. □

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

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

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 approach[2] 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 optimally[3] 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

[2] In 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.

[3] The 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 note be so optimal after all.

problem[4]. 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*

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

where

$$\hat{x}_{t|N} = \hat{x}_{t|t} + S_t(\hat{x}_{t+1|N} - \hat{x}_{t+1|t}), \tag{3.33b}$$
$$P_{t|N} = P_{t|t} + S_t(P_{t+1|N} - P_{t+1|t})S_t^T, \tag{3.33c}$$
$$S_t = P_{t|t}A_t^T P_{t+1|t}^{-1}, \tag{3.33d}$$

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). □

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

[4] In 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-

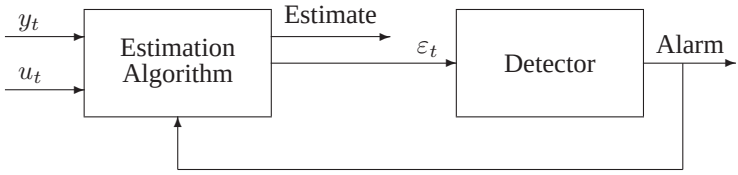

Figure 3.1: *The estimation algorithm delivers residuals ε_t, which are used in the detector to decide whether or not a change has occurred. If a change is detected this information is fed back for use in the estimation algorithm.*

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

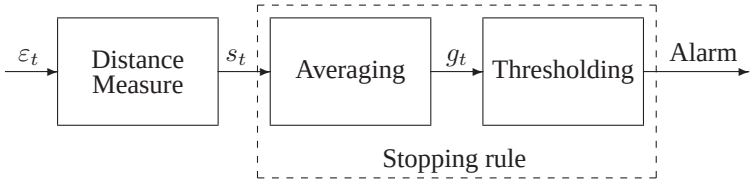

Figure 3.2: *The components of the change detector are a distance measure and a stopping rule, where the latter consists of an averaging and a thresholding procedure.*

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* $t_{\text{alarm}} = t$.
3. *If* $g_t < 0$*:* $g_t = 0$ *and* $\hat{t}_{\text{change}} = t$.

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 $\hat{t}_{\text{change}}$. 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

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

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

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

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

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

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

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

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

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 $\log\left(p(X_t|Y_t)\right)$ 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 software[5] 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

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

where the functions $f_0, \ldots, f_m$ 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

[5] A 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-concave[6]*. 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*

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

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

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

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

[6] A function $f : \mathbf{R}^n \to \mathbf{R}$ 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

$$I\left(g(x_t)\right) \triangleq \mathrm{E}\left\{g(x_t)|Y_s\right\} = \int_{\mathbf{R}^{n_x}} g(x_t)p(x_t|Y_s)\,dx_t. \tag{4.1}$$

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,

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

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 nonlinear 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

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

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

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

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

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

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

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

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,

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

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

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

[1]The 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

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

where

$$q(x^{(i)}) = \frac{t(x^{(i)})}{s(x^{(i)})}, \qquad i = 1, \ldots, M, \tag{4.10}$$

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,

$$\tilde{q}(x^{(i)}) = \frac{q(x^{(i)})}{\sum_{j=1}^{M} q(x^{(j)})}, \qquad i = 1, \ldots, M, \tag{4.11}$$

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

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

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

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

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*

$$q(x^{(i)}) = t(x^{(i)})/s(x^{(i)}), \qquad i = 1, \ldots, M. \tag{4.14}$$

The acceptance probabilities are now obtained by normalization

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

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

$$\Pr\left(\tilde{x}^{(i)} = x^{(j)}\right) = \tilde{q}(x^{(j)}), \qquad i = 1, \ldots, M. \tag{4.16}$$

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

$$t(x) \leq Ls(x), \quad \forall x, \tag{4.17}$$

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* $q(\tilde{x}) = \frac{t(\tilde{x})}{Ls(\tilde{x})}$.
2. *Accept* $\tilde{x}$ *as a sample from* $t(x)$ *with probability* $q(\tilde{x})$*, i.e.,* $\Pr\left(x^{(i)} = \tilde{x}\right) = q(\tilde{x})$. *If* $\tilde{x}$ *is not accepted go back to step* 1.
3. *Repeat step* 1 *and* 2 *for* $i = 1, \ldots, M$.

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*

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

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

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

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

which suggests the following choices

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

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 $\{\tilde{q}^{(i)}\}_{i=1}^{M}$ are calculated according to

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

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.

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

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 $p(y_t|x_{t|t-1})$. 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

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

If this procedure is recursively repeated over time the following approximation

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

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

$$q_t^{(i)} = p\left(y_t | x_{t|t-1}^{(i)}\right), \qquad i = 1, \ldots, M, \tag{4.25}$$

   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*

$$\Pr\left(x_{t|t}^{(i)} = x_{t|t-1}^{(j)}\right) = \tilde{q}_t^{(j)}, \qquad i = 1, \ldots, M. \tag{4.26}$$

4. *Time update: predict new particles according to*

$$x_{t+1|t}^{(i)} \sim p\left(x_{t+1|t} | x_{t|t}^{(i)}\right), \qquad i = 1, \ldots, M. \tag{4.27}$$

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

$$\Pr\left(x_{t|t}^{(i)} = x_{t|t-1}^{(j)}\right) = \tilde{q}_t^{(j)}, \qquad i = 1, \ldots, M. \tag{4.28}$$

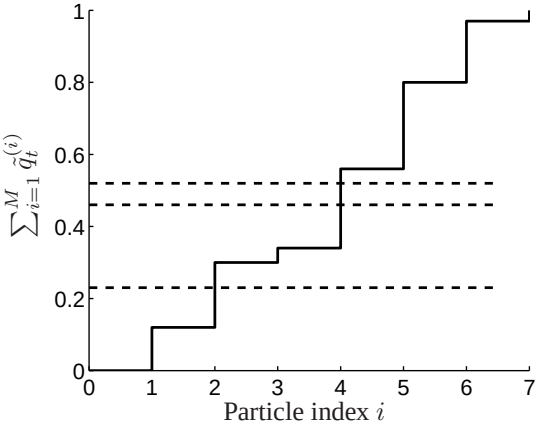

Figure 4.1: *Illustrating the resampling step in the particle filter. The new set of particles is obtained by first generating M sorted uniformly distributed random numbers, three of which are shown by the dashed lines in the figure. These are then associated with a particle guided by the cumulative 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*

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

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

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

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

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

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

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

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,

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

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

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.

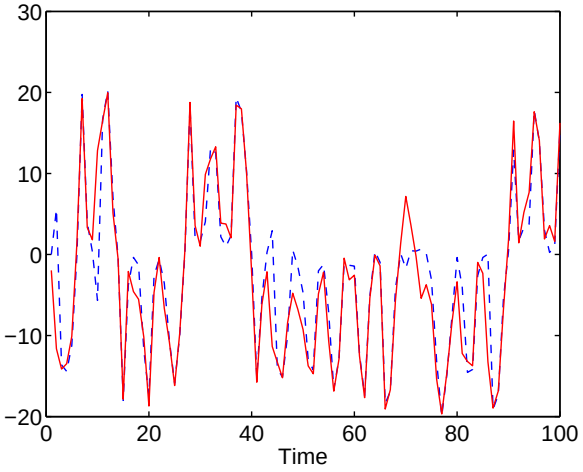

Figure 4.2: *The solid line corresponds to the true state and the dashed line stems from the estimate provided by the particle filter given in Algorithm 4.4. The underlying system is given in (4.33).*

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

$$x_t = \begin{pmatrix} x_t^l \\ x_t^n \end{pmatrix}, \tag{4.34}$$

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

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

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

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

where

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

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)

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

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

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

where

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

Inserting (4.40) into (4.39) gives

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

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

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

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,

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

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* $\{x_{t|t}^{(i)}\}_{i=1}^M$, $t = 1, \ldots, N$.
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*

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

   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*

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

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,

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

This approach has also be 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

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

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

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

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

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

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

$$\mathrm{E}_{p(x_t|Y_s)}\{x_t\} = \int x_t p(x_t|Y_s)\, dx_t. \tag{4.50}$$

Using the following estimate of the probability density function,

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

results in

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

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

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

in (4.1), which after some calculations results in

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

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

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

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

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

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

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

It is often convenient to study the log-likelihood

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

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

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

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

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

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

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 variants[1] (Dennis and Schnabel, 1983). The iterations for the parameter estimates are typically in the following form,

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

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, introduced[2] 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),

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

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,

$$\log p_\theta(Y_N) = \log p_\theta(X_N, Y_N) - \log p_\theta(X_N|Y_N). \tag{5.9}$$

[1] There 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.

[2] The 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 $\theta = \theta'$.

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

where $\mathrm{E}_{\theta'}\{\,\cdot\,|Y_N\}$ 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

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

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

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

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 inequality[3] can be used

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

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

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

$$\mathrm{E}\{f(x)\} \geq f\left(\mathrm{E}\{x\}\right)$$

Algorithm 5.1 (Expectation Maximization (EM))

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

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

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

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

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

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)

$$z_t = \begin{pmatrix} x_t \\ \theta \end{pmatrix}. \tag{5.15}$$

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

$$x_{t+1} = f_1(x_t, u_t, t)\theta_t + f_2(x_t, u_t, t) + w_t, \tag{5.16a}$$
$$\theta_{t+1} = \theta_t + w_t^\theta, \tag{5.16b}$$
$$y_t = h_1(x_t, u_t, t)\theta + h_2(x_t, u_t, t) + e_t, \tag{5.16c}$$

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 where 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 Ninness, 2005), the missing data is defined to be the state sequence, $X_N \triangleq \{x_1, \ldots, x_N\}$. 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 classed 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

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

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,

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

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

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

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

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

The matrix inversion lemma[1] allows us to rewrite (A.6) according to

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

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

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

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

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

The determinants in (A.2) can be written

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

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

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

[1] The 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

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

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

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

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

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

where

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

The time update (3.13b) can be written

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

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

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

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

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

where

$$M_t = P_{t|t} - P_{t|t} A_t (Q_t + A_t P_{t|t} A_t^T)^{-1} A_t P_{t|t}, \tag{A.18a}$$
$$P_{t+1|t} = A_t P_{t|t} A_t^T + Q_t, \tag{A.18b}$$
$$L_t = P_{t|t} A_t^T (Q_t + A_t P_{t|t} A_t^T)^{-1}. \tag{A.18c}$$

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

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

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

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 contol. *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., Christophersen, 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 Autonomus 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 solem 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). Marker-less 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. *Biometrica*, 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., Vacchetti, 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., Elmqvist, 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 Gieseking, 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.