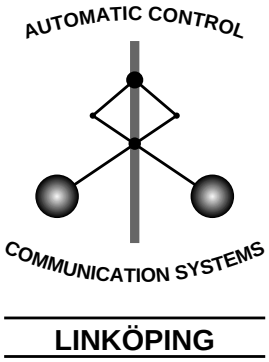
Linköping Studies in Science and Technology
Thesis No. 1047

On Computational Methods for Nonlinear Estimation

Thomas Schön



Division of Automatic Control and Communication Systems
Department of Electrical Engineering
Linköping University, SE–581 83 Linköping, Sweden
WWW: `http://www.control.isy.liu.se`
E-mail: `schon@isy.liu.se`

Linköping 2003

On Computational Methods for Nonlinear Estimation

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

ISBN 91–7373–759–3
ISSN 0280-7971
LiU-TEK-LIC-2003:46

Printed by UniTryck, Linköping, Sweden 2003

To my parents Karin and Stefan,
and my brothers Erik and Sven

Abstract

The Bayesian approach provides a rather powerful framework for handling nonlinear, as well as linear, estimation problems. We can in fact pose a general solution to the nonlinear estimation problem. However, in the general case there does not exist any closed-form solution and we are forced to use approximate techniques. In this thesis we will study one such technique, the sequential Monte Carlo method, commonly referred to as the particle filter. Some work on linear stochastic differential-algebraic equations and constrained estimation using convex optimization will also be presented.

The sequential Monte Carlo method offers a systematic framework for handling estimation of nonlinear systems subject to non-Gaussian noise. Its main drawback is that it requires a lot of computational power. We will use the particle filter both for the nonlinear state estimation problem and the nonlinear system identification problem. The details for the marginalized (Rao-Blackwellized) particle filter applied to a general nonlinear state-space model will also be given.

General approaches to modeling, for instance using object-oriented software, lead to differential-algebraic equations. One of the topics in this thesis is to extend the standard Kalman filtering theory to the class of linear differential-algebraic equations, by showing how to incorporate white noise in this type of equations.

There will also be a discussion on how to use convex optimization for solving the estimation problem. For linear state-space models with Gaussian noise the Kalman filter computes the maximum *a posteriori* estimate. We interpret the Kalman filter as the solution to a convex optimization problem, and show that we can generalize the maximum *a posteriori* state estimator to any noise with log-concave probability density function and any combination of linear equality and convex inequality constraints.

Acknowledgments

First of all I would like to thank my thesis advisor Professor Fredrik Gustafsson, for guiding me in my research. You are a very inspiring advisor, well-equipped with good ideas, and you always find time to discuss my work. I truly enjoy working with you.

I would also like to thank Dr. Jan Maciejowski and Professor Lennart Ljung for introducing me to the world of academic research during my stay at the University of Cambridge in 2001. Without that pleasant stay, this thesis would not exist today.

Several colleges deserve special thanks for always taking their time to listen to my ideas and answering my questions. Especially, I would like to thank the following persons; Dr. Fredrik Tjärnström for helping me out with questions related to system identification and discussing research strategies. Markus Gerdin for a wonderful collaboration with the work on differential-algebraic equations, and for general discussions concerning these equations. Rickard Karlsson for always being helpful when it comes to Kalman and particle filtering issues. Dr. Johan Löfberg for helping me when it comes to actually solving convex optimization problems. Finally, Martin Enqvist for discussing various problems, ranging from system identification to questions on how to write a thesis.

This thesis has been proofread by Dr. Ola Härkegård, Markus Gerdin, Ragnar Wallin, Dr. Fredrik Tjärnström and Rickard Karlsson. Your help has improved the quality of this thesis substantially and I really appreciate that. Any remaining errors are entirely my own. I would also like to thank Ulla Salaneck, who is always helpful when it comes to practical and administrative matters. Gustaf Hendeby has helped me with my constant trouble in getting LaTeX to do what I want, thanks a bunch.

This work has been supported by the Swedish Research Council and the ECSEL graduate school, which is gratefully acknowledged.

I am very grateful to my family for their never-ending support for all that I have undertaken in life this far. Finally, I would like to thank Kristina for all the joy and love you bring into my life.

Thomas Schön

Linköping, October 1, 2003

Contents

1

Introduction

In order to understand how a system of some kind, e.g., a car, an aircraft, a biological system, or a spacecraft, behaves we need to know certain states and parameters associated with the system. Typically we do not have direct access to these, hence they have to be estimated, based on measurements from the system. Moreover, many real systems are mathematically described by nonlinear equations, which motivate the need for nonlinear estimation. We will in this thesis investigate some approaches to nonlinear estimation.

The following section contains a short background and motivation to the work in this thesis. In Section 1.1.1 we discuss stochastic methods and in Section 1.1.2 we discuss deterministic methods. In Section 1.2 we give the outline of the thesis. Finally, in Section 1.3 the main contributions are summarized and a brief story of the research is given.

1.1 Background and Motivation

Let us start with a rather abstract, but hopefully enlightening, description of what this thesis is all about. It is about finding a certain mapping. This mapping takes its arguments from the, typically high-dimensional, space in which the measurements live. This space will be denoted $\mathbb{R}^{t \times n_y}$, where t denotes the current time instant and n_y is the dimension of the measurement vector. The mapping that transforms these measurements into estimates of the states and the parameters we are looking

for, which lives in a space of dimension n_y. We can write this as

$$f_t : \mathbb{R}^{t \times n_y} \rightarrow \mathbb{R}^{n_x}. \tag{1.1}$$

This will provide us with the best estimate of the quantity we are interested in, given the information available in the measurements. At our help in this search we have a model of the underlying system. This model helps us to explore the high-dimensional space in a systematic fashion. Broadly speaking, this thesis is about finding good functions f_t describing how to infer the information available in the measurements on the quantities we are interested in.

We will in this thesis study model based estimation methods. One of the most commonly used model classes within the automatic control and the signal processing community is the state-space model, which consists of a system of possibly nonlinear ordinary differential equations. However, in recent years there has been a trend in modeling to extend this model class to the more general class of models which consists of differential-algebraic equations. The reason is that new object-oriented modeling tools deliver models of this type. One of the topics in this thesis is to extend the standard Kalman filtering theory to the class of linear differential-algebraic equations, by showing how to incorporate white noise in this type of equations.

1.1.1 Stochastic Methods

In our quest of finding the function f_t in (1.1) we will make use of the theory which originated from the work of the English Reverend Thomas Bayes, published two years after his death in Bayes (1763). Today this is referred to as the Bayesian theory. The reason why this theory is so useful is that it provides a systematic method on how to include prior knowledge, typically given by a model, into the estimation procedure. Whether the quantities we are estimating represent the states or the parameters in a dynamical system we will use the Bayesian theory, since it provides a rather powerful framework for handling nonlinear, as well as linear, estimation problems. In fact, we can pose a general solution to the nonlinear estimation problem using Bayes' theory (Jazwinski, 1970). In the linear case this solution can be explicitly written down, resulting in the celebrated Kalman filter introduced by Kalman (1960). However, in the nonlinear case there does not exist any closed-form solution and we are forced to use approximate techniques. A very common idea to tackle this problem is to approximate the nonlinear model by a linear model and then use the optimal Kalman filter for this linearized model. This usually works fine if the nonlinearities are not too severe. However, conceptually speaking this is not a satisfactory solution, since in a way we are solving the wrong problem. If we instead insist on keeping the nonlinear model we can pose the optimal solution. We cannot solve the resulting equations analytically, since they typically involve intractable high-dimensional integrals. We can solve these optimal equations approximately using a class of methods referred to as *sequential Monte Carlo methods*, popularly called particle filters. The first official publication, known to the author, on Monte Carlo methods is Metropolis and Ulam (1949), where the

overall ideas were introduced. However, an important part was missing in order to make the algorithm work properly for our purposes. This part was introduced in Gordon *et al.* (1993). This was the birth of the particle filters. An attractive feature about these methods is, as was noted above, that they provide *an approximate solution to the right problem, and not an optimal solution to the approximate problem*. Furthermore, they provide a systematic framework for handling nonlinear systems, with non-Gaussian noise. The downside is that we require a lot of computational power. The sequential Monte Carlo methods constitute an important part of this thesis. We will use them both for the nonlinear state estimation problem and the nonlinear parameter estimation problem.

1.1.2 Deterministic Methods

Up to this point we have only discussed stochastic approaches to the problem of finding the function f_t. There are also interesting alternatives provided by deterministic methods. In the deterministic setting the estimation problem is basically thought of as a function approximation problem, where we set out to minimize errors. The estimation problem will in the deterministic regime be posed as a convex optimization problem. We will here be able to exploit the fact that it is straightforward to include constraints on the system. Furthermore, the relationship between this deterministic formulation of the optimization problem and the corresponding stochastic formulation will be examined. It is important to realize that methods from the stochastic and the deterministic regime are not to be interpreted as competing methods, rather they complement each other and depending on the problem at hand they will both provide interesting insights. A certain problem might be much easier to tackle using stochastic methods, whereas in another case it might be advantageous to use deterministic methods. In yet other cases they will in fact give the same result if certain assumptions are met.

1.2 Outline

There are two parts in this thesis. In the first part we provide a brief review of the theory necessary to understand the publications in Part II. Another objective with the first part is to explain how the different publications relate to each other and to the existing theory.

1.2.1 Outline of Part 1

Chapter 2 is devoted to stochastic estimation, i.e., when we use stochastic methods to obtain the estimates. There will also be a brief discussion on the different model classes we use. In Chapter 3 we treat the topic of deterministic estimation, within the framework of convex optimization. We will in this chapter also hint at the connections between deterministic and stochastic estimation. Our most important tool for nonlinear estimation in this thesis is the sequential Monte Carlo method,

which are explained in Chapter 4. Finally, we give some concluding remarks in Chapter 5.

1.2.2 Outline of Part 2

This part consists of a collection of four edited papers, briefly described below.

Paper A: 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. Accepted for publication

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.

Paper B: 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

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 we can generalize the MAP state estimator to any noise with log-concave density function and any combination of linear equality and convex inequality constraints on the state.

Paper C: Marginalized Particle Filters for Nonlinear State-space Models

> Schön, T., Gustafsson, F., and Nordlund, P.-J. (2003c). Marginalized particle filters for nonlinear state-space models. *Submitted to IEEE Transactions on Signal Processing*

The recently developed particle filter offers a general numerical tool to approximate the state *a posteriori* density in nonlinear and non-Gaussian filtering problems with arbitrary accuracy. The main drawback with the particle filter is that it is quite computer intensive. For a given filtering accuracy, the computational complexity increases quickly with the state dimension. One remedy to this problem is what in statistics is called Rao-Blackwellization, where states appearing linearly in the dynamics are marginalized out. This leads to that a Kalman filter is attached to each particle. Our main contribution here is to sort out when marginalization is possible for state space models, and to point out the implications in some typical signal processing applications. The methodology and impact in practice is illustrated on terrain navigation for aircraft. The marginalized particle filter for a state-space model with nine states is evaluated on real aircraft data, and the result is that very good accuracy is achieved with quite reasonable complexity.

Paper D: 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

The potential use of the marginalized particle filter for nonlinear system identification is investigated. We derive algorithms for systems which are linear in either the parameters or the states. 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 on 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.

1.3 Main Contributions and Story of the Research

In this section we will explain the main contributions in this thesis, together with a very brief story of the research.

Paper A is the result of work conducted in a very good cooperation with Markus Gerdin, under supervision of Professor Fredrik Gustafsson and Professor Torkel Glad. We showed how to incorporate white noise in linear differential-algebraic equations, resulting in Theorem A.1 and A.3. Using these results we can apply the standard Kalman filtering framework to estimate the internal variables in linear differential-algebraic equations.

The second paper, i.e., Paper B, started out 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 was opponent on the thesis by

Andersson (2002). The main contribution here is that the problem of constrained estimation in linear state-space models was posed as a convex optimization problem.

In Paper C we give the details of the marginalized particle filter for a general nonlinear state-space model. We have extended and improved the results from Nordlund (2002). The example, where the theory is applied to authentic flight data, is the result of the Master's thesis by Frykman (2003), that we supervised.

The final paper, i.e., Paper D explains how the standard and the marginalized particle filter can be applied to the system identification problem. At the ERNSI conference held in Le Croisic in France 2002 someone mentioned that it would be interesting to see whether the particle filter can be useful for the system identification problem or not. This comment, together with the invited session on particle filters held at the 13th IFAC Symposium on System Identification was the reason for conducting the work presented in Paper D.

Part I

Estimation Theory

2

Stochastic Estimation

The topic of this chapter is the state and parameter estimation problem in a stochastic setting. In the subsequent chapter we will study methods for solving the same problem, using a deterministic framework. It is important to realise that methods from the stochastic and the deterministic regime are not to be interpreted as competing methods, rather they complement each other and depending on the problem at hand they will both provide interesting insights. A certain problem might be much easier to tackle using stochastic methods, whereas in another case it might be advantageous to use deterministic methods. In yet other cases they will in fact give the same result if certain assumptions are met. More about this in the next chapter. In Gardner (1986) there is a nice discussion on the relationship between deterministic and stochastic methods based on ergodic theory. Another account of this relationship is provided in the recent book by Kailath *et al.* (2000).

Since we study model based estimation methods in this thesis we will start out by discussing various model classes and how they relate to each other. Specifically, we will discuss how to include stochastics in linear differential-algebraic equations. In Section 2.3 we discuss how these model classes can be used to solve the state estimation problem in a Bayesian framework. Furthermore, we will focus on the problem of recursive estimation in order to set the stage for the sequential Monte Carlo methods introduced in Chapter 4. In Section 2.4 there is a discussion on how the system identification problem is posed in a Bayesian framework, followed by a short review of approximate numerical methods for nonlinear recursive estimation.

2.1 Introduction

The objective in estimation is to obtain information about a certain entity, x_t, given measurements, $Y_s \triangleq \{y_i\}_{i=0}^{s}$ of some kind. This entity has different meanings depending on the context. In this thesis we will mostly be concerned with the discrete-time state estimation problem, i.e., when x_t represents the state of a dynamical system. Besides the discrete-time state estimation problem, we will in Paper D also study the case when x_t includes a parameter vector, i.e., the discrete-time parameter estimation problem, or the system identification problem.

This thesis is about model based estimation methods, i.e., methods that use dynamical models for how the state evolves with time. We need at least two models. The first model describes how the dynamic system behaves, that is how the state evolves over time. It is referred to as the *system model*. The second model describes how the observation is related to the state and is called the *measurement model*.

We will in this thesis take a Bayesian (Bayes, 1763) approach, i.e., we view every unknown entity as a stochastic variable. However, there will also be some comments on the parametric approach according to Fisher (1912) where an unknown entity is thought of as a deterministic variable. In Bayesian estimation the solution to the estimation problem is stated in terms of the *a posteriori* density function for x, $p(x|y)$. This is the distribution of the state, x, given the information provided by the measurement, y. Later in this chapter it will be shown that it is possible to derive an expression for this density. However, it is only in a few special cases that this density can be parameterized using a finite-dimensional description. The most important example is the case with linear dynamics and Gaussian noise. In this case all involved densities will be Gaussian, and hence they can be parameterized using the corresponding mean and covariance. The equations for how these two quantities evolve over time are in this case given by the Kalman filter (Kalman, 1960). In situations that do not lend themselves to finite-dimensional descriptions we can use numerical methods such as the particle filter (Gordon *et al.*, 1993) to obtain an approximate description of the *a posteriori* density. In Chapter 4 we will discuss particle filters in detail. The rest of this chapter is devoted to the problem of stochastic estimation in general, with emphasis on the nonlinear case.

2.2 Model Classes

In our applications models are used to construct various estimates. Our estimation methods rely on a good model of the underlying system. In order to obtain good estimates it is thus crucial that the system at hand has been modeled well enough. This is the reason why we begin this chapter with a discussion on different model classes. We will try to provide a hierarchical classification of the most common model classes, starting with a very general formulation. The discussion will then become more specialized in order to see how this general formulation relates to the model classes commonly used today.

A very general system model is given by the *differential-algebraic equation (DAE)*[1]

$$F(\dot{x}(t), x(t), \theta, w(t), t) = 0. \tag{2.1}$$

The over-dot denotes differentiation with respect to (w.r.t.) time. Moreover, $x \in \mathbb{R}^{n_x}$ is the internal variable vector, $\theta \in \mathbb{R}^{n_\theta}$ is the time-invariant parameter vector, $w \in \mathbb{R}^{n_w}$ is the process noise, and $t \in \mathbb{R}$ denotes time. Finally, the model is described by the possibly nonlinear function $F : \mathbb{R}^{2n_x+n_\theta+n_w+1} \to \mathbb{R}^{n_F}$, where n_F denotes the number of entries in the vector valued function F. We can also have a known input (e.g., a control signal or a measured disturbance) signal, $u(t)$, in (2.1). However, since we in this thesis are concerned with the estimation problem rather than the control problem we will omit the input signal, for reasons of brevity. It is straightforward to include a known input signal when it is present and most of what is discussed in this thesis can easily be adapted to that case.

Studying differential-algebraic equations is interesting because models of this type arise naturally in object-oriented modeling. This way of modeling is becoming increasingly more popular and examples of object-oriented modeling languages are Modelica, Dymola and Omola (Mattsson *et al.*, 1998; Tiller, 2001). This clearly motivates the need to estimate internal variables and parameters in this type of equation. As of today there is no general theory available on how to handle (2.1). 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., Penski, 2000; Römisch and Winkler, 2003; Schein and Denk, 1998; Winkler, 2003). However, a lot still remains to be done within this field. The DAE (2.1) is of stochastic nature. Hence, it is not exactly clear what we mean when we write this equation. It is far from obvious how stochastic processes should be included in this type of equation. If we have white noise in (2.1), there is a risk that derivatives of the white noise will appear in the solution and this is not a well defined process. In Paper A we will show how to properly incorporate white noise in linear stochastic differential-algebraic equations.

Besides the model for how the system behaves we also need a model which describes 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 (we will in the sequel assume that the sampling time is 1 for notational reasons)

$$H(y_k, x_k, \theta, e_k, k) = 0, \tag{2.2}$$

where $y_k \in \mathbb{R}^{n_y}$ is the measurement, $e_k \in \mathbb{R}^{n_e}$ is the measurement noise, $k \in \mathbb{N}$ is the discrete time index, and $H : \mathbb{R}^{n_x+n_\theta+n_e+1} \to \mathbb{R}^{n_H}$ is a possibly nonlinear

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

function describing how the measurements are obtained. We have used n_H to denote the number of entries in the vector valued function H. This kind of implicit measurement function occurs e.g., when we have applications involving map-related measurements, which are used in positioning systems, (see e.g., Hall, 2000; Svenzén, 2002). In Gustafsson *et al.* (2002) map-related measurements are given some attention and they are also related to other measurement models.

2.2.1 A Hierarchical Classification of Models

The previous discussion was deliberately held on a rather abstract level. In this section things will become a bit more concrete. We will show how most of the models used in the signal processing and the automatic control community can be considered to be special cases of the rather general formulation in terms of differential-algebraic equations. Four different model classes are presented in a nested fashion. Model 4 is a special case of Model 3, which is a special case of Model 2, which in turn is a special case of Model 1. There are of course many different ways in which this classification can be done. We have chosen to do this in a way that we believe serves our purposes best.

If we use (2.1) and (2.2) we can formulate a quite general model class, the DAE model.

Model 1 (DAE model)

The nonlinear stochastic differential-algebraic equation (DAE) model is given by

$$F(\dot{x}(t), x(t), \theta, w(t), t) = 0, \tag{2.3a}$$

$$H(y_k, x_k, \theta, e_k, k) = 0, \tag{2.3b}$$

where $w(t)$ and e_k are assumed to be white noises, i.e.,

$$p(w(t), w(s)) = p(w(t))p(w(s)), \qquad \forall t \neq s, \qquad \text{where } t, s \in \mathbb{R}, \tag{2.3c}$$

$$p(e_k, e_l) = p(e_k)p(e_l), \qquad \forall k \neq l, \qquad \text{where } k, l \in \mathbb{N}. \tag{2.3d}$$

Furthermore the Jacobian $\partial F/\partial \dot{x}$ can be singular.

The mathematically inclined will probably object to the noise definition used here and suggest that we use Itô calculus instead (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 though. The rest of this section is devoted to describing the most important special cases.

An important special case of Model 1 arise when the Jacobian w.r.t. $\dot{x}$ is nonsingular, i.e., $\partial F/\partial \dot{x} \neq 0$, and $\dot{x}(t)$ can be explicitly solved for. The resulting model is then given by the ordinary differential equation (ODE) model, commonly referred to as the continuous-time *state-space model*. Hence, the system behavior is governed by ordinary differential equations rather than by differential-algebraic equations. To conform with the existing literature we will in this special case refer to the internal variable as the state. The ODE model is defined below.

Model 2 (ODE model)

The nonlinear stochastic ordinary differential equation (ODE) model is given by

$$\dot{x}(t) = f(x(t), \theta, w(t), t), \tag{2.4a}$$
$$y_k = h(x_k, \theta, e_k, k), \tag{2.4b}$$

where $w(t)$ and e_k are assumed to be white noises, i.e.,

$$p(w(t), w(s)) = p(w(t))p(w(s)), \qquad \forall t \neq s, \qquad \text{where } t, s \in \mathbb{R}, \tag{2.4c}$$
$$p(e_k, e_l) = p(e_k)p(e_l), \qquad \forall k \neq l, \qquad \text{where } k, l \in \mathbb{N}. \tag{2.4d}$$

In Model 2 the noise terms enter the equations in a nonlinear fashion. In order to make the mathematics simpler it is often assumed that the noise instead enters additively in (2.4a) – (2.4b). Furthermore, we will in this thesis be concerned with discrete-time estimation problems, and hence we need a discrete-time model. See e.g., Gustafsson (2000) for a discussion on how to discretize a continuous-time nonlinear model. The following discrete-time state-space model, and several of its special cases, is the model class most widely used in the literature on discrete-time state estimation.

Model 3 (Discrete-time nonlinear state-space model with additive noise)

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

$$x_{t+1} = f(x_t, \theta, t) + w_t, \tag{2.5a}$$
$$y_t = h(x_t, \theta, t) + e_t, \tag{2.5b}$$

where w_t and e_k are assumed to be white noises, i.e.,

$$p(w_t, w_s) = p(w_t)p(w_s), \qquad \forall t \neq s, \qquad \text{where } t, s \in \mathbb{N}, \tag{2.5c}$$
$$p(e_t, e_s) = p(e_t)p(e_s), \qquad \forall t \neq s, \qquad \text{where } t, s \in \mathbb{N}. \tag{2.5d}$$

In Model 3 defined above we have used t to index the discrete time. This convention is used, when there is no risk of confusion, in order to get a more consistent notation. It is worth noting that it is far from clear how the discretization of the state noise should be performed. In the literature there exist several different ideas on how to do this by imposing different assumptions on the state noise. In Gustafsson (2000, Section 8.9.4) several different ideas on how to perform this discretization are discussed.

An important special case of Model 3 is when all equations are linear, i.e.,

Model 4 (Discrete-time linear state-space model)

The discrete-time linear state-space model is given by

$$x_{t+1} = A_t x_t + B_t w_t, \tag{2.6a}$$
$$y_t = C_t x_t + e_t, \tag{2.6b}$$

where w_t and e_t are assumed to be white Gaussian noises, i.e.,

$$\mathrm{E}\left[w_t w_s^T\right] = Q_t \delta_{ts}, \qquad \mathrm{E}\left[e_t e_s^T\right] = R_t \delta_{ts}, \qquad \forall t, s, \qquad \text{where } t, s \in \mathbb{N}. \tag{2.6c}$$

In Model 4 above δ_{ts} is Kronecker's delta function, which is 0 whenever $t \neq s$, and 1 when $t = s$. This model can of course also have parameters present, which have to be estimated along with, or separately from the state. In that case the parameters enter the model in the A_t, B_t, C_t, Q_t, and R_t matrices. The problem of simultaneously estimating the states and the parameters is studied in Paper D. It is worth mentioning that the B_t matrix in (2.6a) can equally well be included in the covariance matrix for $\tilde{w}_t$, where $\tilde{w}_t = B_t w_t$. We write (2.6a) as $x_{t+1} = A_t x_t + \tilde{w}_t$, and the covariance of $\tilde{w}_t$ is $\mathrm{E}[\tilde{w}_t \tilde{w}_t^T] = B_t Q_t B_s^T \delta_{ts}$. Furthermore, it is worth noting that if we use the same noise source in both (2.6a) and (2.6b) and use the Kalman gain matrix instead of B_t we obtain the innovation form (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 discussed in Ljung (1999).

Since there currently is a trend in modeling to use the more general Model 1 it is necessary to extend the estimation theory to cope with systems described by differential-algebraic equations. A first step in this direction is to study linear DAE:s, which is done in the subsequent section and in Schön *et al.* (2003a).

2.2.2 Linear Differential-algebraic Equations

In this thesis, Model 3 and some of its special cases will mostly be used. However, we will also study extensions to differential-algebraic equations. As mentioned above, the first obstacle to overcome is how to introduce stochastics into this type of equation. This will be discussed in this section and in Paper A a proposal is given for how to introduce white noise, and how to estimate the internal variables in linear differential-algebraic equations, i.e., Model 5, defined below. More specifically we will discuss what properties the white noise has to posses in order for Model 5 to be well defined.

Model 5 (Linear differential-algebraic equation (DAE))

The linear differential-algebraic equation (DAE) model is given by

$$E\dot{x}(t) + Fx(t) = B_w w(t), \tag{2.7a}$$
$$y_k = Cx_k + e_k, \tag{2.7b}$$

where E might be singular and w_t and e_k are white Gaussian noises, i.e.,

$$\mathrm{E}\left[w_t w_s^T\right] = Q_t\delta(t-s), \qquad \forall t, s, \qquad \text{where } t, s \in \mathbb{R}, \tag{2.7c}$$
$$\mathrm{E}\left[e_k e_l^T\right] = R_k\delta_{kl}, \qquad \forall k, l, \qquad \text{where } k, l \in \mathbb{N}. \tag{2.7d}$$

In Model 5 above $\delta(\cdot)$ is Dirac's delta function, with the following important property

$$\int_{-\infty}^{\infty} g(s)\delta(s-t)ds = g(t). \tag{2.8}$$

There are two main reasons for why we want to introduce white noise in linear differential-algebraic equations:

- There are unmodeled dynamics and disturbances acting on the system. They can be included in the model as an unknown stochastic term.
- There is a practical need for tuning the filter in order to make a trade-off between tracking ability and sensor noise attenuation. In the Kalman filter, this can accomplished by keeping the sensor noise covariance matrix constant and tuning the process noise covariance matrix, or the other way around. Often, it is easier to describe the sensor noise in a stochastic setting. Then it is more natural to tune the state noise.

When we know how to incorporate white noise into linear differential-algebraic equations we have taken the first step towards being able to use standard statistical methods in order to estimate internal variables and parameters in this model class. The state estimation problem is discussed in Paper A, (Schön *et al.*, 2003a) and the parameter estimation problem is discussed in Gerdin *et al.* (2003). In the discrete time case much work has already been done, (see e.g., Dai, 1987, 1989; Darouach *et al.*, 1993; Deng and Liu, 1999; Nikoukhah *et al.*, 1998, 1999). However, the models obtained from object-oriented modeling languages are mostly in continuous time and hence we have to be able to introduce noise in the continuous-time models as well.

The problem when it comes to introducing white noise in DAE:s is that derivatives of white noise might affect the internal variables directly. The physical interpretation of this is that it would require infinite forces, currents etc. This is exemplified in Paper A. In order to avoid this problem we will derive a basis for

the subspace of all disturbances that leads to a causal system. This basis is taken as B_w in (2.7a), and the process noise covariance matrix $Q_t = \mathrm{Cov}[w_t]$ is used as the design variable to rotate and scale this basis.

For linear differential-algebraic equations there exists an important standard form according to the following theorem.

Theorem 2.1 (Standard form for Model 5)
Suppose that there exists a λ such that $\lambda E + F$ is invertible. Then there exist nonsingular matrices P, Q such that (2.7) can be written as (Q is used as a variable substitution, $z(t) = Qx(t)$ and P is multiplied from the left in (2.7a))

$$\begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} + \begin{bmatrix} -A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} w(t), \tag{2.9}$$

where N is a matrix of nilpotency k, i.e., $N^k = 0$ for some k.

Proof Kronecker's canonical form (see Gantmacher, 1959; Kailath, 1980) provides a proof for the existence of this standard form. □

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 ideas from Varga (1992) involving the generalized real Schur form and the generalized Sylvester equation the problem of numerically computing the standard form (2.9) can be handled. Furthermore, we do not have to calculate the exact Kronecker canonical form, where the A, and N matrices in (2.9) have to be on Jordan form.

If we rewrite (2.9) according to

$$\dot{z}_1(t) = Az_1(t) + G_1 w(t), \tag{2.10a}$$

$$z_2(t) = \sum_{i=0}^{k-1} (-N)^i G_2 \frac{d^i w(t)}{dt^i}, \tag{2.10b}$$

we see that we can prevent white noise from being differentiated by requiring that

$$NG_2 = 0. \tag{2.11}$$

In Paper A we utilize this in order to derive the subspace which corresponds to that we do not differentiate white noise. We also derive this subspace using frequency-domain methods. Using the subspace derived with the time-domain result we can discretize (2.10) and estimate the internal variables by means of the standard Kalman filter. The reader is referred to Paper A for the details.

2.3 Bayesian State Estimation

When it comes to stochastic estimation problems there are two main paradigms, the Bayesian and the Fisherian. The Bayesian paradigm originates from the English Reverend Thomas Bayes, who stated Bayes' theorem, which was published two years after his death in Bayes (1763). The Fisherian paradigm derives from the work of Fisher (1912). The main difference is that in the Bayesian approach both the state and the measurement are thought of as random variables, whereas in the Fisherian approach the state is thought of as fixed, but unknown, hence it is *not* treated as a random variable. The state and the measurement are related by the *likelihood* function, $p(y|x)$. Within the Fisherian paradigm the likelihood function is often written as $l(x)$ to emphasize that it is regarded as a function of the state, x, after the observation has been inserted. Let us from now on focus on the Bayesian paradigm.

As previously mentioned the objective in Bayesian estimation is to use information from observations of certain random variables (the measurement) in order to infer information on another random variable (the state). The random vector x is assumed to have an *a priori* density function, $p(x)$, which contains all the information about the state prior to the experiment is performed and the measurement is obtained. When the measurement is obtained we need information about how it is related to the state. This relation is provided by the likelihood function. In words, the likelihood function provides information about how *likely* the measurement is given the state. The likelihood function is provided by the measurement model (2.2). Exactly how the likelihood function is obtained from the measurement model is explained in Section 2.3.2.

The information obtained from the measurement is inferred on the state by using Bayes' theorem Bayes (1763),

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}, \tag{2.12}$$

where the denominator is just a positive scalar constant, which can be thought of as a normalizing constant. It can be obtained by marginalization according to

$$p(y) = \int_{\mathbb{R}^{n_x}} p(x, y)dx = \int_{\mathbb{R}^{n_x}} p(y|x)p(x)dx. \tag{2.13}$$

A useful property in discussing Bayesian estimation is the *Markov property*, which says that the state, x_t, contains all information available about the system at time t. This property is sometimes referred to as the generalized causality principle: the future can be predicted from knowledge of the present (Jazwinski, 1970). With a Bayesian approach, the most general solution to the estimation problem is provided by the *a posteriori* density, given by (2.12). Given the information about the state before the experiment is performed, $p(x)$, and the measurement, y, obtained after the experiment is conducted, *all information* about the state is now available in the *a posteriori* density $p(x|y)$. When we have obtained the *a posteriori* density,

which indeed can be hard, we can compute the optimal[2] point estimate for any loss function. A discussion on various loss functions can be found in Jazwinski (1970), and in the following section we will discuss some of the most important point estimates. Besides providing various point estimates the *a posteriori* density can be used e.g., to calculate confidence intervals. An early discussion on the use of the Bayesian approach in stochastic estimation and control problems is given in Ho and Lee (1964).

2.3.1 Estimators and Estimates

As mentioned above the complete solution to the estimation problem is given by the *a posteriori* density function, $p(x|y)$. However, we are often interested in a point estimate of the unknown state, x. This is provided by an *estimator*, $\hat{x}$. An estimator is a function of a random variable, and hence it is itself a random variable,

$$\hat{x} = g(y). \tag{2.14}$$

When we insert a measurement, y, in the estimator we obtain a realization of the estimator, i.e., an *estimate*, $\hat{x}$.

There are several different ways of choosing the estimator, some more obvious than others. We will here discuss some of the most common estimators. See e.g., (Anderson and Moore, 1979) and (Kailath *et al.*, 2000) for more information about estimators. In Jazwinski (1970) there is a deeper discussion on how to choose the estimator.

The most obvious estimate is perhaps the most probable outcome,

$$\boxed{\hat{x}^{MAP} = \arg\max_x p(x|y) = \arg\max_x p(y|x)p(x)} \tag{2.15}$$

which is referred to as the maximum *a posteriori* (MAP) estimate. In (2.15) we have in the second equality used Bayes' theorem (2.12) together with the fact that the maximization is performed over x.

Another common estimate is the *maximum likelihood* (ML) estimate introduced by Fisher (1912),

$$\boxed{\hat{x}^{ML} = \arg\max_x p(y|x)} \tag{2.16}$$

By comparing (2.15) and (2.16) we can deduce that the only difference between the MAP and the ML estimate is the term $p(x)$. This difference derives from the fact that in a Bayesian framework all variables are thought of as random.

[2]These estimates are optimal in the sense that they are the solution to a certain optimization problem. This means that it is important that the optimization problem is properly posed, otherwise the optimal estimate might not be so optimal after all.

Besides finding an estimate, we also need to assess the quality of the estimate. One way of doing this is to use the measure

$$\mathrm{E}\left[\|x-\hat{x}\|^2 |y\right] = \int_{\mathbb{R}^{n_x}} \|x-\hat{x}(y)\|^2 p(x|y)dx, \tag{2.17}$$

which also can be obtained from the *a posteriori* density, $p(x|y)$. This quality measure suggests another type of estimator, the *minimum variance* (MV) estimator. It is obtained by minimizing (2.17). The minimum variance estimate is given by

$$\boxed{\hat{x}^{MV} = \mathrm{E}\left[x|y\right]} \tag{2.18}$$

or in words, the conditional expectation of the state given the measurement (Anderson and Moore, 1979).

2.3.2 Recursive State Estimation

This section is devoted to recursive state estimation. Two good surveys on nonlinear recursive estimation are provided by (Sorenson, 1988) and (Kulhavý, 1996). Another good reference for nonlinear recursive and nonrecursive estimation is (Jazwinski, 1970). For linear recursive estimation (Kailath *et al.*, 2000) and (Anderson and Moore, 1979) contain a lot of information. In discussing recursive estimation problems the notation $\hat{x}_{t|t-1}$ is useful. It means, the estimate of the state x_t at time t, given the information available at time $t-1$. Another useful notation is Y_t, which is used to denote all measurements up to time t, i.e., $\{y_i\}_{i=0}^{t}$. Recursive estimation is about obtaining an estimate of the current state, x_t, given information about the last estimate and the current measurement, y_t. Using the notation introduced above this means that we want to find $\hat{x}_{t|t-1}$. That is, we want a recursive scheme for how to update the *a posteriori* density recursively as new measurements arrive. There exists a well known solution for how to perform this update. We give it here in the following theorem, which can be used for Model 3 and special cases thereof (Jazwinski, 1970).

Theorem 2.2 (Discrete-time Bayesian recursive estimation)
The filtering, $p(x_t|Y_t)$, and the one step prediction, $p(x_{t+1}|Y_t)$, densities are recursively given by a measurement update according to

$$p(x_t|Y_t) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})}, \tag{2.19a}$$

$$p(y_t|Y_{t-1}) = \int_{\mathbb{R}^{n_x}} p(y_t|x_t)p(x_t|Y_{t-1})dx_t, \tag{2.19b}$$

and a time update according to

$$p(x_{t+1}|Y_t) = \int_{\mathbb{R}^{n_x}} p(x_{t+1}|x_t)p(x_t|Y_t)dx_t, \tag{2.20}$$

and the recursion is initiated by $p(x_0|Y_{-1}) = p(x_0)$.

Proof In this proof Bayes' theorem and the Markov property are used, which follows the development in Jazwinski (1970).
Measurement update:

$$p(x_t|Y_t) = \frac{p(Y_t|x_t)p(x_t)}{p(Y_t)} = \frac{p(y_t, Y_{t-1}|x_t)p(x_t)}{p(y_t, Y_{t-1})} \tag{2.21a}$$

$$= \frac{p(y_t|Y_{t-1}, x_t)p(Y_{t-1}|x_t)p(x_t)}{p(y_t|Y_{t-1})p(Y_{t-1})} \tag{2.21b}$$

$$= \frac{p(y_t|x_t)p(x_t|Y_{t-1})p(Y_{t-1})p(x_t)}{p(y_t|Y_{t-1})p(Y_{t-1})p(x_t)} = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \tag{2.21c}$$

Moreover, using Bayes' theorem and the Markov property we obtain

$$p(y_t, x_t|Y_{t-1}) = p(y_t|x_t, Y_{t-1})p(x_t|Y_{t-1}) = p(y_t|x_t)p(x_t|Y_{t-1}), \tag{2.22}$$

which in turns implies that (by integrating w.r.t. x_t on both sides)

$$p(y_t|Y_{t-1}) = \int_{\mathbb{R}^{n_x}} p(y_t|x_t)p(x_t|Y_{t-1})dx_t. \tag{2.23}$$

Time update:

$$p(x_{t+1}, x_t|Y_t) = p(x_{t+1}|x_t, Y_t)p(x_t|Y_t) = p(x_{t+1}|x_t)p(x_t|Y_t), \tag{2.24}$$

where we have used Bayes' theorem in the first equality and the Markov property in the second. We obtain the time update equation (another name for this equation is the *Chapman-Kolmogorov* equation) by integrating both sides w.r.t. x_t, i.e.,

$$p(x_{t+1}|Y_t) = \int_{\mathbb{R}^{n_x}} p(x_{t+1}, x_t|Y_t)dx_t = \int_{\mathbb{R}^{n_x}} p(x_{t+1}|x_t)p(x_t|Y_t)dx_t. \tag{2.25}$$

□

There is however 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 dynamical model is linear, i.e., Model 4, and when the noises and the initial condition are both normally distributed. Since all the involved densities are Gaussian and we only perform linear operations, the state will at all times be Gaussian as well. The solution is provided by the Kalman filter (Kalman, 1960).

Theorem 2.3 (Kalman filter)
Consider Model 4, and assume that the noise and the initial state are Gaussian, i.e., $w_t \sim \mathcal{N}(0, Q_t), e_t \sim \mathcal{N}(0, R_t), x_0 \sim \mathcal{N}(\bar{x}_0, \bar{P}_0)$. *Then, the filter estimate and the prediction estimate are also normally distributed according to*

$$x_t|Y_t \sim \mathcal{N}(\hat{x}_{t|t}, P_{t|t}), \tag{2.26a}$$

$$x_{t+1}|Y_t \sim \mathcal{N}(\hat{x}_{t+1|t}, P_{t+1|t}), \tag{2.26b}$$

where

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - C_t\hat{x}_{t|t-1}), \tag{2.27a}$$
$$K_t = P_{t|t-1}C_t^T S_t^{-1}, \tag{2.27b}$$
$$S_t = C_t P_{t|t-1} C_t^T + R_t, \tag{2.27c}$$
$$P_{t|t} = P_{t|t-1} - P_{t|t-1}C_t^T S_t^{-1} C_t P_{t|t-1}, \tag{2.27d}$$
$$\hat{x}_{t+1|t} = A_t\hat{x}_{t|t}, \tag{2.27e}$$
$$P_{t+1|t} = A_t P_{t|t} A_t^T + B_t Q_t B_t^T, \tag{2.27f}$$

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

Proof It can be proved by using Theorem 2.2 and imposing the assumptions of a linear Model 4, Gaussian noise, and Gaussian initial conditions. This is rather straightforward, however tedious, work. See e.g., Nordlund (2002) for details. For other ways of proving the Kalman filter the reader is referred to e.g., Kailath *et al.* (2000), Anderson and Moore (1979), Gustafsson (2000). □

When no analytical solution exists we are forced to use some approximate method. Some of these are briefly mentioned in Section 2.5 and in Chapter 4 we discuss one family of these methods in more depth, the sequential Monte Carlo methods.

In Theorem 2.2 the result is given in terms of probability density functions (PDF:s), but the model classes in Section 2.2 are not given in terms of PDF:s. However, it is possible to express the model classes using PDF:s as well. We will here explain how Model 3 can be translated into a description using PDF:s. If we assume that x_t is known the only stochastic part in (2.5a) is the state noise, w_t. Hence, we have that

$$p(x_{t+1}|x_t) = p_{w_t}(x_{t+1} - f(x_t, \theta, t)). \tag{2.28}$$

Furthermore, if x_t is known the only stochastic part in (2.5b) is the measurement noise, e_t, and we have that

$$p(y_t|x_t) = p_{e_t}(y_t - h(x_t, t)). \tag{2.29}$$

For the more advanced model classes, where the noise does not enter the equations linearly, Theorem 2.7 in Jazwinski (1970, page 34) can be used.

2.4 Bayesian System Identification

In this section we show that taking a Bayesian view of the system identification problem will basically transform the problem into the Bayesian state estimation problem. A nice treatment of the system identification problem in a Bayesian context is given in Peterka (1979) and in Kramer and Sorenson (1988). In the latter reference a simple and enlightening example is discussed and the authors

stress that the Bayesian approach provides a powerful alternative viewpoint of the parameter estimation problem. Using the Bayesian methods we estimate the *a posteriori* density, not just a few of its first moments, as is typically the case with point estimators.

Consider the problem of estimating the parameter vector, θ, in Model 3. For the simple special case, where the parameters are linear in the observations and the noise is white and Gaussian, there exists an explicit solution to the problem. This will result in the least squares method. However, in the general case estimation of the parameter vector in Model 3 is a very hard problem.

The idea commonly used in the system identification literature is to augment the state, z_t, with the parameter vector, θ, according to Ljung and Söderström (1983).

$$\bar{x}_t = \begin{bmatrix} x_t \\ \theta \end{bmatrix} \tag{2.30}$$

and consider the state estimation problem for the augmented state, $\bar{x}_t$. This will result in a new system, which still falls in the class of models given by Model 3, according to

$$\bar{x}_{t+1} = \begin{bmatrix} f(x_t, \theta, t) \\ \theta \end{bmatrix} + \begin{bmatrix} w_t \\ 0 \end{bmatrix}, \tag{2.31a}$$

$$y_t = h(x_t, \theta, t) + e_t. \tag{2.31b}$$

Hence we are faced with the nonlinear estimation problem, which we have discussed in the previous sections and the solution is provided by Theorem 2.2. However, these equations do not have an explicit solution except in a few special cases. In order to handle the general nonlinear case we have to use approximate numerical methods, which will be briefly discussed in the subsequent section. In Paper D, i.e., (Schön and Gustafsson, 2003) we have discussed the nonlinear system identification problem using the augmentation (2.30) and sequential Monte Carlo methods. A recent approach to the system identification problem using the Markov chain Monte Carlo (MCMC) method is given in Ninness and Henriksen (2003).

Now, recall that with the Bayesian view on statistics the true parameter is assumed to be a random variable. Therefore, the Bayesian framework is well suited for the case when the parameter is time-varying. So far we have only discussed the case of a constant, i.e., time-invariant parameter. If the parameter is allowed to be time-varying, e.g., according to a simple random walk,

$$\theta_{t+1} = \theta_t + w_t^\theta, \tag{2.32}$$

we are directly faced with the state estimation problem. In this case the parameter is in fact a state. To elaborate a bit, the only difference between the state, x_t, and the time-varying parameter, θ_t (2.32) is that they obey different dynamical equations. Hence, there is in fact no structural difference between the state and the time-varying parameter.

2.5 Approximate Numerical Methods

Most of the problems we encounter in practice are of a nonlinear nature, but as mentioned above there does not exist an analytical solution to the truly nonlinear recursive estimation problem[3]. This implies that we are forced to approximations of some kind in order to approach this problem. The approximations suggested in literature this far are of two different types, either they approximate the model by a simpler one or they use numerical methods to find a global approximation of the *a posteriori* density (Kulhavý, 1996; Sorenson, 1974). These two alternatives are briefly discussed below.

2.5.1 Model Approximations

The most commonly used strategy is to approximate the model. This is typically done by expanding the nonlinear functions around the "working point" at every time step, using Taylor series. Then a model which fits the linear Model 4 is obtained by truncation of the Taylor series. Furthermore the noise processes involved are approximated by Gaussian distributions. We now have a linear model with Gaussian noise, and we know that the Kalman filter is the optimal solution for estimating the state in this case. The filter obtained by approaching the nonlinear estimation problem in this manner is referred to as the *extended Kalman filter (EKF)* (Anderson and Moore, 1979; Kailath *et al.*, 2000). Another way of approximating the model is to force the state vector to belong to a finite set of values. This model class is referred to as a *hidden Markov model (HMM)* (Elliott *et al.*, 1995).

2.5.2 Global Approximation of the A Posteriori Density

We know that the solution to the nonlinear recursive estimation problem is given by Theorem 2.2. That theorem is neglected in methods using model approximation. However, if we choose to use this theorem we can use the nonlinear models as we have derived them from the underlying physics and approximate the optimal solution provided by Theorem 2.2 by using numerical methods. Over the years several different methods for performing this approximation have appeared. We will mention a few of the most important ones. For more references (see e.g., Bergman, 1999; Kulhavý, 1996).

If we approximate the *a posteriori* density using a sum of Gaussian densities according to

$$p(x_t|Y_t) \approx \sum_{i=1}^{N} q_t^{(i)} \mathcal{N}(\hat{x}_{t|t}^{(i)}, P_{t|t}^{(i)}), \qquad \sum_{i=1}^{N} q_t^{(i)} = 1 \tag{2.33}$$

we obtain the *Gaussian sum* approach (Alspach and Sorenson, 1972; Sorenson and Alspach, 1971). Another approximation is provided by the *point-mass filter*

[3]There are a few pathological exceptions from this statement.

(Bergman, 1999; Bucy and Senne, 1971) which, as the name reveals, means that the *a posteriori* density is approximated by a set of points on a predefined grid,

$$p(x_t|Y_t) \approx \sum_{i=1}^{N} q_t^{(i)} \delta(x_t - x_t^{(i)}), \qquad \sum_{i=1}^{N} q_t^{(i)} = 1 \tag{2.34}$$

Another approach, which can be interpreted as an extension of the point-mass filter is the *sequential Monte Carlo method*, also referred to as *particle filter* (Doucet, 1998; Doucet *et al.*, 2001a; Gordon *et al.*, 1993). In these algorithms the *a posteriori* density is also approximated by a set of points, however the grid is chosen in a stochastic, rather than in a deterministic manner, as is the case in point-mass filters. In Chapter 4 the sequential Monte Carlo method is discussed in more detail.

3

Deterministic Estimation

This chapter is devoted to the estimation problem without the stochastic framework used in the previous chapter. Hence, the estimation problem is treated as a deterministic problem of minimizing errors. One of the advantages of studying the estimation problem in a deterministic setting is that it is straightforward to utilize prior information about the state, such as constraints. The reason why it is so is that we formulate the state estimation problem as an optimization problem. A nice historical account of the relationship between the Kalman filter, i.e., stochastic methods, and the corresponding deterministic optimization problem is given in Sorenson (1970).

The unconstrained estimation problem is discussed in Section 3.1, where we pose the estimation problem in a deterministic framework. This is followed by a discussion on the stochastic interpretation of the deterministic least squares method. In Section 3.3 we discuss state estimation using convex optimization, which provides a systematic way to incorporate constraints in the estimation procedure.

3.1 Unconstrained Estimation

In this section the stochastic assumptions are removed from the discrete-time linear Model 4 introduced in Chapter 2. Hence, we will think of w_t and e_t simply as errors of unknown character. Given a set of measurements, $Y_t \triangleq \{y_i\}_{i=0}^{t}$, and a guess of the initial state x_0, which we label $\bar{x}_0$, we are faced with a problem of curve fitting.

That is, we want to determine the state in Model 4 in such a way that it describes the actual measurements as well as possible. That is, 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 problem

$$\boxed{\begin{array}{lrcll} \min\limits_{X_t} & \|x_0 - \bar{x}_0\|^2_{\bar{P}_0^{-1}} + \sum_{i=0}^{t-1} \|w_i\|^2_{Q_i^{-1}} + \sum_{i=0}^{t} \|e_i\|^2_{R_i^{-1}} & & & \\ \text{s.t.} & x_{i+1} & = & A_i x_i + w_i, & i = 0, \ldots, t-1 \\ & y_i & = & C_i x_i + e_i, & i = 0, \ldots, t \end{array}} \tag{3.1}$$

where the weighting matrices $\{Q_i\}_{i=0}^{t}$, $\{R_i\}_{i=0}^{t}$ and $\bar{P}_0$ are design parameters. This is a convex problem, in fact it is a quadratic program (QP).

As time increases, the number of variables in (3.1) will increase. Hence, we need to bound the number of variables in some way. One way to do this is to derive a recursive solution. If we impose a stochastic interpretation to the problem this will again result in the Kalman filter. See (Rao, 2000) for a derivation. The Kalman filter is in other words the recursive solution to the weighted least squares problem (3.1). We will return to this issue in the subsequent section. Another way of bounding the number of variables is to segment the measurements and use a *moving horizon strategy*. That is, first we solve the least squares problem using the measurements Y_t. As a new measurement appears we solve the same problem again. However, this time we use the measurements $Y_{1:t+1} = \{y_i\}_{i=1}^{t+1}$, and so on. This is in fact the dual problem to the unconstrained *model predictive control* (MPC) problem (Goodwin, 2003). We will return to this duality later in Section 3.3, when we extend the problem and introduce constraints as well.

3.2 Stochastic Interpretation

In this section we will give (3.1) a stochastic interpretation. We start out by deriving the maximum *a posteriori* estimate, i.e.,

$$\boxed{\hat{X}_t = \arg\max_{X_t} p(X_t|Y_t)} \tag{3.2}$$

with respect to the more general Model 3. The derivation follows Cox (1964) and Jazwinski (1970).

According to Bayes' theorem we have

$$p(X_t|Y_t) = \frac{p(Y_t|X_t)p(X_t)}{p(Y_t)} \tag{3.3}$$

We can discard the denominator in (3.3), since $p(X_t|Y_t)$ is minimized w.r.t. X_t and the denominator is clearly independent of X_t. The likelihood function $p(y_t|x_t)$ can be written as

$$p(y_t|x_t) = p_{e_t}(y_t - h(x_t)), \tag{3.4}$$

since we assumed additive noise in Model 3. We also have $p(x_{t+1}|x_t) = p_{w_t}(x_{t+1} - f(x_t))$. Now, since the stochastic sequences w_t and e_t are both assumed to be independent we obtain

$$p(Y_t|X_t) = \prod_{i=0}^{t} p(y_i|x_i) = \prod_{i=0}^{t} p_{e_i}(y_i - h(x_i)) \tag{3.5}$$

$$p(X_t) = p_{x_0}(x_0) \prod_{i=0}^{t-1} p(x_{i+1}|x_i) = p_{x_0}(x_0) \prod_{i=0}^{t-1} p_{w_i}(x_{i+1} - f(x_i)) \tag{3.6}$$

Putting it all together we arrive at

$$p(X_t|Y_t) = c p_{x_0}(x_0) \prod_{i=0}^{t-1} p_{w_i}(x_{i+1} - f(x_i)) \prod_{i=1}^{t} p_{e_i}(y_i - h(x_i)), \tag{3.7}$$

where $c \in \mathbb{R}^+$ derives from $p(Y_t)$. $p(X_t|Y_t)$ can be used as the objective function for the optimization problem. Due to the fact that the logarithmic function is strictly monotone we may equally well consider

$$\begin{aligned} \log(p(X_t|Y_t)) \propto \log(p_{x_0}(x_0)) &+ \sum_{i=0}^{t-1} \log(p_{w_i}(x_{i+1} - f(x_i))) \\ &+ \sum_{i=0}^{t} \log(p_{e_i}(y_i - h(x_i))), \end{aligned} \tag{3.8}$$

where we have dropped the positive constant c, since it does not affect the solution to the optimization problem. We now see that (3.2) can be written as

$$\boxed{\begin{array}{lrcll} \max\limits_{X_t} & \log(p_{x_0}(x_0)) + \sum_{i=0}^{t-1} \log(p_{w_i}(w_i)) + \sum_{i=0}^{t} \log(p_{e_i}(e_i)) & & & \\ \text{s.t.} & x_{i+1} & = & f(x_i) + w_i, & i = 0, \ldots, t-1 \\ & y_i & = & h(x_i) + e_i, & i = 0, \ldots, t \end{array}} \tag{3.9}$$

which is a convex problem if p_{x_0}, p_{w_i} and p_{e_i} are log-concave[1] and f and h are both linear functions.

If we now impose the assumptions that the model is linear and that w_i and e_i are normally distributed we obtain

$$\begin{aligned} \log(p(X_t|Y_t)) \propto \log\left(e^{-\|x_0 - \bar{x}_0\|^2_{\bar{P}_0^{-1}}}\right) &+ \sum_{i=0}^{t-1} \log\left(e^{-\|x_{i+1} - A_i x_i\|^2_{Q_i^{-1}}}\right) \\ &+ \sum_{i=0}^{t} \log\left(e^{-\|y_i - C_i x_i\|^2_{R_i^{-1}}}\right), \end{aligned} \tag{3.10}$$

[1] A function $f : \mathbb{R}^n \to \mathbb{R}$ is *log-concave* if $f(x) > 0$ for all x in the domain of f and $\log(f)$ is a concave function (Vandenberghe and Boyd, 2001).

which implies that (3.2) in this case can be written as

$$\boxed{\max_{X_t} \quad \|x_0 - \bar{x}_0\|^2_{\bar{P}_0^{-1}} + \sum_{i=0}^{t-1} \|x_{i+1} - A_i x_i\|^2_{Q_i^{-1}} + \sum_{i=0}^{t} \|y_i - C_i x_i\|^2_{R_i^{-1}}} \tag{3.11}$$

This is precisely the optimization problem, to which the Kalman filter is the solution. When we solve (3.11) we will besides the filter estimate, $\hat{x}_{t|t}$, also obtain the smoothed estimates, $\hat{x}_{k|t}$, where $k < t$. If we only are interested in the filtered estimate we should have considered the problem

$$\boxed{\hat{x}_{t|t} = \arg\max_{x_t} p(x_t|Y_t)} \tag{3.12}$$

instead. If we now compare the two optimization problems (3.1) and (3.11), we see that they indeed are *equivalent*, if we interpret the weighting matrices in (3.1) as covariance matrices. It is also worthwhile to note the subtle difference between these two problems. In (3.11) the constraints are inserted into the objective function, whereas in (3.1) they are explicitly kept as constraints in the formulation. (3.1) and (3.11) are equivalent problems, and this illustrates the fact that in the statistical literature constraints are made implicit by inclusion in the objective function and hence avoided. In Section (3.3) we will discuss what happens if we allow ourselves to keep the constraints explicit as well as adding more constraints to the problem. The addition of new constraints will open new possibilities in the estimation procedure, which is not possible if we insist on making all the constraints implicit.

It is worth noting that the solution to (3.1) contains many of the well known estimation algorithms as special cases. This will be exemplify here using the linear regression model, which is a very common model class in applications (Gustafsson, 2000).

$$y_t = \varphi_t^T \theta_t + e_t, \tag{3.13}$$

where φ_t is the regression vector and θ_t is the state to be estimated. There are several standard methods to estimate the state in this type of model, e.g., Least Squares (LS), Recursive Least Squares (RLS) with or without forgetting factor, λ, and Windowed Least Squares (WLS) (Gustafsson, 2000). In order to see this we first note that if the linear regression model (3.13) is interpreted as the measurement equation in a state-space model,

$$\theta_{t+1} = \theta_t + w_t, \qquad \mathrm{Cov}[w_t] = Q_t \tag{3.14a}$$

$$y_t = \varphi_t \theta_t + e_t, \qquad \mathrm{Cov}[e_t] = R_t. \tag{3.14b}$$

we have exactly the model class used in (3.1). Now, the different estimation methods mentioned above can be interpreted as different assumptions on the weighting (i.e., covariance) matrices Q_t and R_t.

3.3 Convex Optimization Estimation

This section is devoted to the problem of formulating estimation problems as convex optimization problems. The advantage of casting the estimation problem as a convex optimization problem is that we can easily add certain constraints to the problem and obtain the optimal solution using standard software. In this way we can utilize prior information about the state, e.g., that the state is always positive and that the components of the state sum to one, as is the case if the state is a vector of probabilities. Constraints of this type cannot be 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 (3.1), it is straightforward to include the constraints. We will here briefly introduce the ideas. For a more complete treatment the reader is referred to Paper B, i.e., (Schön *et al.*, 2003b), where we also provide an example on estimating probabilities. Performing state estimation using optimization techniques has previously been discussed using quadratic programs in e.g., Rao *et al.* (2001), Rao (2000), and Robertson and Lee (2002). For a nice introduction to constrained estimation and its connection to model predictive control (MPC) (Maciejowski, 2002) the reader is referred to (Goodwin, 2003). Both these problems are treated at a more technical level in Michalska and Mayne (1995).

Before starting we give a very brief account of convex optimization. The main message of convex optimization is that we should *not* differ between linear and nonlinear optimization problems, but instead between convex and nonconvex problems. The class of convex problems is much larger than that covered by linear problems, and we know that for a convex problem any local optimum is also the global optimum. Moreover, there exists efficient algorithms for solving convex optimization problems. A convex optimization problem is defined as

$$\boxed{\begin{array}{lrcll} \min\limits_{x} & f_0(x) & & & \\ \text{s.t.} & f_i(x) & \leq & 0, & i=0,\ldots,m \\ & a_i^T x & = & b_i, & i=0,\ldots,n \end{array}} \tag{3.15}$$

where the functions $f_0, \ldots, f_m$ are convex and the equality constraints are linear. The x-variable is in this general formulation the optimization variable, not to be mistaken for the state of a dynamical system. For a thorough introduction to convex optimization the reader is referred to (Vandenberghe and Boyd, 2001).

We will in this section be concerned with the MAP estimate (2.15), according to

$$\boxed{\hat{z}^{MAP} = \arg\max_{z} \ \log(p(z|y))} \tag{3.16}$$

The z-variable is a nuisance variable, whereas the x-variable is typically interpreted as the state variable of a dynamical system if we are studying the state estimation problem. If we are concerned with the system identification problem the x-variable will be the parameter vector. In Paper B both the state estimation and the system

identification problem are considered. Now, assume that we want to estimate $[x^T, z^T]^T$, where z has a certain known distribution, and that x and z are related through the linear constraints

$$A \begin{bmatrix} x \\ z \end{bmatrix} = b. \tag{3.17}$$

If we now want to use (3.16), we are faced with the problem of finding the joint distribution of x and z, which can be quite tedious. This can be avoided by considering the following convex optimization problem instead

Problem 1 (Convex optimization estimation)

Assume that $p(z|y)$ is a known log-concave probability density function. The MAP-estimate for $[x^T, z^T]^T$, where x and z are related via (3.17) is given by

$$\begin{aligned} \max_{x,z} \quad & \log(p(z|y)) \\ \text{s.t.} \quad & A \begin{bmatrix} x \\ z \end{bmatrix} = b, \end{aligned}$$

It is here worth noting that any linear equalities and convex inequalities may be added to this formulation, and standard software applies. In the subsequent section we will make this somewhat abstract statement a bit more concrete by considering the filtering problem.

3.3.1 Convex Optimization Filtering

In the previous section we talked about constraints in general. This section is devoted to a special type of constraints, namely the ones that appear in describing the dynamic behavior of a system. In order to obtain convex problems we will use linear models of the dynamics, that is Model 4. However, in Paper B we have used an extended version of Model 4, which includes descriptor systems and systems on innovation form as well.

To express the state filtering problem in the general estimation formulation given in Problem 1, let

$$z = \begin{bmatrix} x_0 \\ W_{t-1} \\ E_t \end{bmatrix} \qquad \text{and} \qquad x = X_t. \tag{3.18}$$

One objective function that can be used is (3.9), and the constraints are given by Model 4. This results in the following optimization problem

Problem 2 (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. In the presence of constraints in terms of a linear dynamic Model 4, the MAP-estimate is the solution $\hat{x}_t = x_t$ to the following problem

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

The same remark given for the general estimation Problem 1 is still valid, i.e., that any linear equalities and convex inequalities may be added to this formulation, and we can use standard software for the resulting problem. It is also worth stressing that it is straightforward to include other variables to be estimated, such as e.g., missing data into Problem 2. Besides including them in the variables to be estimated there is probably also a need to give some assumptions on how they behave, which are typically implemented as constraints.

Another type of constraints that might be interesting to add to Problem 2 are those that makes it possible to include model uncertainty. Let us assume that we are uncertain about the A-matrix in Problem 2, one way of expressing this is to say that the A-matrix should belong to a set of some kind. Depending on the properties of this set we will obtain different optimization problems. This is in the literature referred to as *robust estimation*. For information about commonly used sets and the resulting optimization problems and how to solve them, see e.g., Ghaoui and Lebret (1997), and Vandenberghe and Boyd (2001).

For several of the most common noise densities it is possible to implement Problem 2 by rewriting it as a *second-order cone program (SOCP)* Lobo *et al.* (1998), which is a convex optimization problem where a linear function is minimized over the intersection of second-order cones and an affine set. In the problem definition below, the x-variable is to be thought of as an optimization variable, not as the state of a dynamical system.

Problem 3 (Second-order cone program (SOCP))

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad \|A_i x + b_i\|_2 &\leq c_i^T x + d_i, \qquad i = 1, \ldots, N \\ Fx &= g \end{aligned}$$

The problem parameters are $c, c_i \in \mathbb{R}^n, A_i \in \mathbb{R}^{n_i \times n}, b_i \in \mathbb{R}^{n_i}, d_i \in \mathbb{R}, F \in \mathbb{R}^{p \times n}$, and $g \in \mathbb{R}^p$. The optimization variable is $x \in \mathbb{R}^n$.

We will not go into the details of this rewriting Problem 2 as a SOCP. However, once it is done it is straightforward to implement it using YALMIP (Löfberg, 2003), which is a useful and intuitive interface between MATLAB and state of the art solvers for semidefinite optimization problems (of which SOCP is a special case).

The main concern with the formulation of the estimation problem given in Problem 2 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 2 with additional constraints.

Another way of bounding the number of variables in the optimization problem is to use *moving horizon estimation (MHE)*. This is basically the same idea underpinning model predictive control (MPC), i.e., we estimate the state using a fixed size, moving window of data. This idea was used in the windowed least squares approach previously discussed in Section 3.2. More concrete, when we estimate the state at the next time instant we remove the oldest measurement and include the new measurement. This will hence result in the problem defined below.

Problem 4 (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

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

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

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 suboptimal, since the influence of the past measurements is not taken care of correctly.

The formulation used in Problem 4 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. The extension to nonlinear systems discussed below can also be useful for the change detection problem. In such an extension the solution would probably be based on ideas similar to the innovation whiteness test of the filter bank approach discussed in Gustafsson (2000, Chapters 8 and 9).

Problem 4 can be extended to the nonlinear estimation problem, by using the nonlinear Model 3 instead of the linear Model 4. Even though the extension is very

simple the problem now becomes much harder, since the optimization becomes nonconvex. Rao *et al.* (2001) and Rao (2000) provide good entry points into the literature on moving horizon estimation for nonlinear systems.

4

Sequential Monte Carlo Methods

As was explained in Chapter 2, the complete solution to the Bayesian estimation problem is provided by the *a posteriori* density, $p(X_t|Y_t)$. This density contains all information available about the state $X_t \triangleq \{x_i\}_{i=0}^t$. The objective is thus to estimate, recursively in time, the *a posteriori* density and its associated features such as the *filtering density* $p(x_t|Y_t)$ and expectation on the form

$$I(g(X_t)) = \mathrm{E}_{p(X_t|Y_t)}\left[g(X_t)\right] = \int g(X_t)p(X_t|Y_t)dX_t. \tag{4.1}$$

In the Kalman filter, where it is assumed that the system is linear and subject to Gaussian noise, it can be proved that the *a posteriori* density is also Gaussian. Hence the *a posteriori* density belongs to a function class that can be parameterized using two parameters (the mean and the covariance). This means that is sufficient to update these two parameters. However, in the general case it is very hard, and in most cases indeed impossible, to find a function class to parameterize the *a posteriori* density. The central idea in sequential Monte Carlo methods is to use a set of random samples, with associated weights, to represent the *a posteriori* density instead of a parameterized function class. In this way it is possible to consider nonlinear and non-Gaussian problems in a systematic fashion.

The sequential Monte Carlo methods belong to a larger class of algorithms referred to as Monte Carlo methods (Andrieu *et al.*, 2003). An account of the whole spectrum of Monte Carlo methods is given in (Andrieu *et al.*, 2001). We will now give a very short review of the history of the Monte Carlo method. For a

more thorough review the reader is referred to Eckhardt (1987). The first public document published on the Monte Carlo method is Metropolis and Ulam (1949). This paper introduces several of the ideas that form the basis of modern sequential Monte Carlo methods. The next improvement was the Metropolis algorithm, published in Metropolis *et al.* (1953). This algorithm was improved in Hastings (1970) resulting in the Metropolis-Hastings algorithm. In 1990 the Monte Carlo methods made their first major impact on the statistical community, mainly due to the great increase in computational power (Andrieu *et al.*, 2001). The birth of the sequential Monte Carlo method, i.e., the particle filter was in 1993 through the seminal paper by Gordon *et al.* (1993). The literature on the particle filter is by now quite extensive and two good references are (Doucet *et al.*, 2001a) and (Doucet, 1998).

The discussion in this chapter applies to Model 3, previously defined in Section 2.2.1 and its special cases. In Section 4.1 we assume that it is possible to sample from the *a posteriori* density. However, since this is in general not the case Section 4.2 is devoted to importance sampling, which is a method to solve this problem. Furthermore, it is in this section described how to make the solution recursive. After these introductory sections we are ready to formulate the particle filter algorithm, which we do in Section 4.3. There will then be a discussion on how to improve the estimates using a variance reduction method based on marginalization of probability density functions. The chapter is concluded by a discussion on applications of the particle filter.

4.1 Perfect Sampling

To begin with, let us assume that it is possible to draw N independent and identically distributed (i.i.d.) samples (also called particles, hence the name particle filter), $\{X_t^{(i)}\}_{i=1}^N$ from $p(X_t|Y_t)$. An empirical estimate of the *a posteriori* density is then given by (Doucet *et al.*, 2001a)

$$\hat{p}_N(X_t|Y_t) = \frac{1}{N}\sum_{i=1}^{N}\delta(X_t - X_t^{(i)}), \tag{4.2}$$

where $\delta(\cdot)$ is Dirac's delta function defined in Section 2.2.2. Using this empirical density we now obtain an estimate of $I(g(X_t))$ according to

$$\hat{I}_N(g(X_t)) = \int g(X_t)\hat{p}_N(X_t|Y_t)dX_t = \frac{1}{N}\sum_{i=1}^{N} g(X_t^{(i)}). \tag{4.3}$$

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

$$\hat{I}_N(g(X_t)) \xrightarrow{\text{a.s.}} I(g(X_t)) \quad \text{as} \quad N \to \infty, \tag{4.4}$$

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

$$\sqrt{N}\left[\hat{I}_N(g(X_t)) - I(g(X_t))\right] \xrightarrow{d} \mathcal{N}(0, \sigma_t^2) \quad \text{as} \quad N \to \infty, \tag{4.5}$$

where $\xrightarrow{d}$ denotes convergence in distribution (Doucet *et al.*, 2001a).

The problem is that we cannot sample directly from the *a posteriori* density, $p(X_t|Y_t)$, due to its complex nature. A popular method to obtain samples from complex densities is the Markov Chain Monte Carlo (MCMC) method (Robert and Casella, 1999). This method is based on the simulation of a Markov chain, whose limiting density is the *a posteriori* density. The problem is that this method is iterative, and hence not applicable to recursive problems. Hence, we need another method which lends itself to recursive implementation. One popular solution is provided by the importance sampling framework (Doucet *et al.*, 2000). This will be the topic of the subsequent section.

4.2 Importance Sampling

The idea behind importance sampling is to sample from a density that is simple to sample from, the *importance density*[1], $\pi(X_t|Y_t)$, instead of the more complex *a posteriori* density. The only assumption imposed on the importance density is that its support includes the support of $p(X_t|Y_t)$, otherwise we risk division by zero. It is clear that we can write (4.1) according to

$$I(g(X_t)) = \int g(X_t)\frac{p(X_t|Y_t)}{\pi(X_t|Y_t)}\pi(X_t|Y_t)dX_t = \mathrm{E}_{\pi(X_t|Y_t)}\left[g(X_t)q(X_t)\right], \tag{4.6}$$

where the *importance weight*, $q(X_t)$, is defined as

$$q(X_t) = \frac{p(X_t|Y_t)}{\pi(X_t|Y_t)}. \tag{4.7}$$

The numerator in (4.7) can be expressed as

$$p(X_t|Y_t) = \frac{p(Y_t|X_t)p(X_t)}{p(Y_t)}. \tag{4.8}$$

The problem is that typically there does not exist any closed-form expression for the normalization constant $p(Y_t)$. Hence, the importance weight can only be evaluated up to a normalizing constant, which can be solved by studying

$$I(g(X_t)) = \frac{\int g(X_t)q(X_t)\pi(X_t|Y_t)dX_t}{\int q(X_t)\pi(X_t|Y_t)dX_t}, \tag{4.9}$$

[1]Other common names for this density are proposal density and importance function (Bergman, 1999).

instead of (4.6). Note, the in (4.9) we have used the fact the

$$\int q(X_t)\pi(X_t|Y_t)dX_t = \int p(X_t|Y_t)dX_t = 1. \tag{4.10}$$

We then obtain normalized importance weights, as will be described below. If we sample N i.i.d. samples $\{X_t^{(i)}\}_{i=1}^N$ according to $\pi(X_t|Y_t)$ we have (compare with (4.2))

$$\pi(X_t|Y_t) \approx \frac{1}{N}\sum_{i=1}^{N}\delta(X_t - X_t^{(i)}) \tag{4.11}$$

and a Monte Carlo estimate of $I(g(X_t))$ in (4.9) is provided by

$$\hat{I}_N(g(X_t)) = \frac{\frac{1}{N}\sum_{i=1}^{N} q(X_t^{(i)})g(X_t^{(i)})}{\frac{1}{N}\sum_{i=1}^{N} q(X_t^{(i)})} = \sum_{i=1}^{N}\tilde{q}_t^{(i)}g(X_t^{(i)}), \tag{4.12}$$

where the *normalized importance weights* are defined as

$$\tilde{q}_t^{(i)} = \tilde{q}(X_t^{(i)}) = \frac{q(X_t^{(i)})}{\sum_{j=1}^{N} q(X_t^{(j)})} \tag{4.13}$$

The importance sampling method just explained is thus equivalent to using the following approximation of the *a posteriori* density

$$\hat{p}_N(X_t|Y_t) = \sum_{i=1}^{N}\tilde{q}_t^{(i)}\delta(X_t - X_t^{(i)}). \tag{4.14}$$

It is worth noting that if we could sample from the *a posteriori* density, i.e., $\pi(X_t|Y_t) = p(X_t|Y_t)$, we will obtain $\tilde{q}^{(i)} = 1/N, \forall\, i$, which in turn implies that we are back in the perfect sampling framework discussed in the previous section. The strong law of large numbers applies to the estimate (4.12), i.e., $\hat{I}_N(g(X_t)) \xrightarrow{\text{a.s.}} I(g(X_t))$ as $N \to \infty$. By invoking certain other technical assumptions the central limit theorem holds as well (Doucet *et al.*, 2001a; Geweke, 1989). The estimate will thus have nice asymptotic properties.

4.2.1 Sequential Importance Sampling

The preceding discussions have been of batch nature, now it is time to study how to make these results recursive. In the case of importance sampling all the importance weights have to be recalculated as a new measurement becomes available. The aim here is thus to derive an estimate of $p(X_t|Y_t)$, using the estimate of $p(X_{t-1}|Y_{t-1})$ and the new measurement, y_t. In other words, we want to obtain the new samples $\{x_t^{(i)}\}_{i=1}^N$ and the corresponding weights $\{\tilde{q}_t^{(i)}\}_{i=1}^N$ using the old realizations,

$\{X_{t-1}^{(i)}\}_{i=1}^N$, the old weights, $\{\tilde{q}_{t-1}^{(i)}\}_{i=1}^N$, and the new measurement, y_t. The key step is to assume that the importance density is on the form[2] (Doucet *et al.*, 2001a)

$$\pi(X_t|Y_t) = \pi(x_0)\prod_{i=1}^{t}\pi(x_i|X_{i-1}, Y_i), \tag{4.15}$$

which allows us to evaluate the importance weights recursively in time as new measurements becomes available. In order to derive the update equations for the weights we start by expressing $p(X_t|Y_t)$ in terms of $p(X_{t-1}|Y_{t-1})$, $p(y_t|x_t)$, and $p(x_t|x_{t-1})$. Repeated use of Bayes' theorem and the Markov property, both defined in Section 2.3, gives

$$\begin{aligned} p(X_t|Y_t) = p(X_t|Y_{t-1}, y_t) &= \frac{p(X_t, y_t|Y_{t-1})}{p(y_t|Y_{t-1})} \\ &\propto p(y_t|X_t, Y_{t-1})p(X_t|Y_{t-1}) \\ &= p(y_t|x_t)p(x_t|X_{t-1}, Y_{t-1})p(X_{t-1}|Y_{t-1}) \\ &= p(y_t|x_t)p(x_t|x_{t-1})p(X_{t-1}|Y_{t-1}) \end{aligned} \tag{4.16}$$

Now, inserting (4.16) and

$$\pi(X_t|Y_t) = \pi(x_t|X_{t-1}, Y_t)\pi(X_{t-1}|Y_{t-1}) \tag{4.17}$$

in (4.7) gives the following equation for recursively updating the weights

$$q_t^{(i)} = \frac{p(X_t|Y_t)}{\pi(X_t|Y_t)} \tag{4.18}$$

$$\propto \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})p(X_{t-1}|Y_{t-1})}{\pi(x_t^{(i)}|X_{t-1}^{(i)}, Y_t)\pi(X_{t-1}|Y_{t-1})} = \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{\pi(x_t^{(i)}|X_{t-1}^{(i)}, Y_t)}q_{t-1}^{(i)}. \tag{4.19}$$

Moreover, it is commonly assumed that $\pi(x_t|X_{t-1}, Y_t) = \pi(x_t|x_{t-1}, y_t)$, which implies that when we are estimating the filtering density we only need to store $\{x_t^{(i)}\}_{i=1}^N$, and not the entire history $\{X_{t-1}^{(i)}\}_{i=1}^N$ and Y_{t-1}. The final weight update equation will thus be

$$q_t^{(i)} \propto \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{\pi(x_t^{(i)}|x_{t-1}^{(i)}, y_t)}q_{t-1}^{(i)} \tag{4.20}$$

The sequential importance sampling (SIS) is summarized in Algorithm 4.1, based on Doucet *et al.* (2000).

[2]Sequential importance sampling is thus a special case of importance sampling, where the importance densities are forced to be on the form (4.15).

Algorithm 4.1 (Sequential importance sampling (SIS))

1. *For* $i = 1, \ldots, N$, *sample* $x_t^{(i)} \sim \pi(x_t|X_{t-1}^{(i)}, Y_t)$ *and set* $X_t^{(i)} = (X_{t-1}^{(i)}, x_t^{(i)})$.
2. *For* $i = 1, \ldots, N$, *assign the importance weights, up to a normalizing constant, according to*

$$q_t^{(i)} = \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{\pi(x_t^{(i)}|X_{t-1}^{(i)}, Y_t)} q_{t-1}^{(i)} \tag{4.21}$$

3. *For* $i = 1, \ldots, N$, *normalize the importance weights*

$$\tilde{q}_t^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^{N} q_t^{(j)}} \tag{4.22}$$

A special case of this algorithm was introduced by Handschin and Mayne (1969) and Handschin (1970). Many of the algorithms proposed in this field have later been shown to be special cases of this general algorithm (Arulampalam *et al.*, 2002).

In the subsequent section we will connect the theory of recursive estimation from Section 2.3.2 with the theory presented in this chapter, in order to obtain a method to approximate the optimal solution to the recursive filtering problem.

4.3 The Particle Filter

When it comes to recursive Bayesian estimation we are generally most interested in one of the marginal densities of the *a posteriori* density, the *filtering density*

$$p(x_t|Y_t) = \int_{\mathbb{R}^{n_x}} \cdots \int_{\mathbb{R}^{n_x}} p(x_0, \ldots, x_t|Y_t) dx_0 \ldots dx_{t-1} \tag{4.23}$$

$$= \int_{(\mathbb{R}^{n_x})^{t-1}} p(X_t|Y_t) dX_{t-1}, \tag{4.24}$$

or some similar prediction density $p(x_{t+k}|Y_t), k \in \mathbb{N}$. According to Theorem 2.2 the filtering density can be obtained by recursively updating the following densities

$$p(x_t|Y_t) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})}, \tag{4.25a}$$

$$p(y_t|Y_{t-1}) = \int_{\mathbb{R}^{n_x}} p(y_t|x_t)p(x_t|Y_{t-1}) dx_t, \tag{4.25b}$$

$$p(x_{t+1}|Y_t) = \int_{\mathbb{R}^{n_x}} p(x_{t+1}|x_t)p(x_t|Y_t) dx_t. \tag{4.25c}$$

The two integrals (4.25b) and (4.25c) are both on the form (4.1) and hence we can use sequential importance sampling in order to obtain recursive approximations of them.

There is however a severe problem inherent in the sequential importance sampling method that we have still not addressed. That is that the quality of the importance weights, and hence the quality of the estimated density, will deteriorate with time. In fact, it has been shown that the variance of the importance weights will increase with time and hence the estimate will diverge for all importance densities on the form (4.15) (Kong *et al.*, 1994). The solution to this problem is the resampling step, which roughly means that particles having small importance weights are discarded and particles having large importance weights are multiplied. Inspired by the work on the weighted bootstrap by Smith and Gelfand (1992), the resampling step was introduced in the paper by Gordon *et al.* (1993). This was the birth of the particle filter. The divergence problem can also be limited by using better choice of the importance density. Below the resampling idea is discussed and later in this chapter different importance densities are briefly discussed.

4.3.1 The Resampling Step

The main idea behind the resampling step is to discard particles with small weights, i.e., small likelihood, and to multiply particles with large weights, i.e., particles corresponding to large likelihoods. This is done by drawing a new set of particles, $\{x_{t|t}^{(i)}\}_{i=1}^{N}$, with replacement from the old particles, $\{x_{t|t-1}^{(i)}\}_{i=1}^{N}$, according to

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

This can be thought of as the measurement update in the particle filter, since we now infer the information available in the measurement on the state. The reason for using the notation $x_{t+1|t}$ and $x_{t|t}$ is to make it easier to follow the particle filter algorithm and its function. For more details on the resampling step and its implementation we refer to (Bergman, 1999).

The resampling step does indeed reduce the divergence problem, however it also introduce other problems. Theoretically, it introduces a dependence between the different particles, which makes convergence results harder to establish. Moreover, particles having large weights will be selected many times, which gives rise to the problem known as *sample impoverishment* (Arulampalam *et al.*, 2002). This problem gives a loss in diversity among the particles. It arises due to the fact that in the resampling stage the samples are drawn from a discrete PDF, rather than from a continuous one. In the literature 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 and to make the particles differ more from each other. Another idea to reduce the sample impoverishment problem is to resample from continuous approximations of the discrete PDF. This is referred to as the *regularized particle filter*. It would

be interesting to investigate if there is some kind of formal connection between the roughening procedure and the regularized particle filter.

Taking these negative effects of the resampling step into account we would like to avoid resampling if it is not necessary. In order to do this we need a measure of the degeneracy of the algorithm. One such measure is provided by the effective sample size, N_{eff}, defined as (Bergman, 1999)

$$N_{\text{eff}} = \frac{N}{1 + \text{Var}_{\pi(\cdot|Y_t)}\left[q_t\right]} = \frac{N}{\text{E}_{\pi(\cdot|Y_t)}\left[(q_t)^2\right]}, \tag{4.27}$$

where q_t is the importance weight, i.e., $q_t = p(x_t|Y_t)/\pi(x_t|x_{t-1}, y_t)$. In the second equality above we have used the fact that $\text{Var}[q_t] = \text{E}[(q_t)^2] - (\text{E}[q_t])^2$. This cannot be evaluated exactly, but an approximation is provided by

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (\tilde{q}_t^{(i)})^2}, \tag{4.28}$$

where $\{\tilde{q}_t^{(i)}\}_{i=1}^N$ are the normalized importance weights. If the variance of the importance weights is large we will have a small $\hat{N}_{\text{eff}}$ according to (4.28). When $\hat{N}_{\text{eff}}$ is smaller than a certain user defined threshold, N_{th}, we apply the resampling step in order to decrease the variance of the importance weights. Directly after the resampling step we have $\hat{N}_{\text{eff}} = N$. For more information on the efficient sample size see e.g., Bergman (1999). Below we give the algorithm for a generic particle filter for Model 3 in accordance with Arulampalam *et al.* (2002).

Algorithm 4.2 (Generic particle filter for Model 3)

1. *Initialization: For $i = 1, \ldots, N$, initialize the particles according to $x_{1|0}^{(i)} \sim p_{x_0}(x_0)$.*
2. *Measurement update 1: For $i = 1, \ldots, N$, evaluate the importance weights $\{q_t^i\}_{i=1}^N$ according to (4.20) and normalize, i.e., $\tilde{q}_t^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^N q_t^{(j)}}$.*
3. *Calculate the efficient sample size*

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (\tilde{q}_t^{(i)})^2} \tag{4.29}$$

4. *Measurement update 2: If $\hat{N}_{\text{eff}} < N_{th}$ resample with replacement N particles according to*

$$\Pr(x_{t|t}^{(i)} = x_{t|t-1}^{(j)}) = \tilde{q}_t^{(j)}. \tag{4.30}$$

   and reset the importance weights, $q_t^{(i)} = \frac{1}{N}, \forall i$.
5. *Time update: For $i = 1, \ldots, N$ predict new particles according to*

$$x_{t+1|t}^{(i)} \sim p(x_{t+1|t}|x_{t|t}^{(i)}, y_t). \tag{4.31}$$

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

In order to get an intuitive understanding for the above algorithm it is useful to think about the particle filter as a simulation based estimation method, i.e., we simulate a large number of possible state trajectories. Each of these trajectories is assigned a weight according to the likelihood function. The resampling step will update the trajectories according to the weights, so that only the most likely ones are used in the sequel. Evaluating the importance weights and performing the resampling can be thought of as the *measurement update* in the particle filter. It is at these stages that the information in the measurement is inferred on the state. The trajectories are then predicted using the dynamics describing the system. This is the *time update*. These predicted particles will then form the starting point of another iteration of the algorithm.

Before we move on, some of the most important advantages and disadvantages with the particle filter are stated. First the disadvantages,

- There is a lack of convergence results, but there is work going on in this direction (Crisan and Doucet, 2002).
- It is a computer intensive method.

The most important advantages are,

- It can be applied to a very general class of nonlinear, non-Gaussian estimation problems.
- It provides an approximation of the entire *a posteriori* density, not just a single point estimate.
- It is easy to implement, as will be illustrated in Section 4.3.4.
- It works in practice. See e.g., Doucet *et al.* (2001a) and Gustafsson *et al.* (2002) for several application examples.

4.3.2 Obtaining the Estimates

As described above the particle filter provides us with an estimate of the filtering density, $p(x_t|Y_t)$, from which we can deduce several point estimates, e.g., according to (4.1). An estimate of the mean value of the current state can be obtained by inserting $g(x_t) = x_t$ in

$$I(g(x_t)) = \mathrm{E}_{p(x_t|Y_t)}\left[g(x_t)\right] = \int g(x_t)p(x_t|Y_t)dx_t, \tag{4.32}$$

which is a marginalized version of (4.1). This yields

$$I(x_t) = \mathrm{E}_{p(x_t|Y_t)}[x_t] = \int x_t p(x_t|Y_t)dx_t. \tag{4.33}$$

Using the estimate of the *a posteriori* density,

$$\hat{p}_N(x_t|Y_t) = \sum_{i=1}^{N} \tilde{q}_t^{(i)} \delta(x_t - x_t^{(i)}), \tag{4.34}$$

provided by the particle filter, gives

$$\hat{x}_{t|t} = \int x_t \hat{p}_N(x_t|Y_t) dx_t = \int x_t \sum_{i=1}^{N} \tilde{q}_t^{(i)} \delta(x_t - x_t^{(i)}) dx_t = \sum_{i=1}^{N} \tilde{q}_t^{(i)} x_t^{(i)}. \tag{4.35}$$

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

$$g(x_t) = (x_t - \hat{x}_{t|t})(x_t - \hat{x}_{t|t})^T \tag{4.36}$$

in (4.32), which after some calculations result in

$$\hat{P}_{t|t} = \sum_{i=1}^{N} \tilde{q}_t^{(i)} (x_t^{(i)} - \hat{x}_{t|t})(x_t^{(i)} - \hat{x}_{t|t})^T. \tag{4.37}$$

From the two expressions (4.35) and (4.37) it is evident 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.

4.3.3 Design Parameters

In order to obtain some structure in the by now vast literature on particle filters it is instructive to differ between the model, the algorithm, and the design parameters. When it comes to linear systems with Gaussian noise we use the Kalman filter in order to solve the estimation problem. In this case the design parameters are the initial state $\bar{x}_0$, its covariance, $\bar{P}_0$, the state noise covariance Q_t, and the measurement noise covariance R_t.

In estimating the state in nonlinear systems with non-Gaussian noise using the particle filter the design parameters are much more complicated. In fact it is not really clear which the design parameters are. Everyone will agree that the number of particles, N, and the PDF for the initial state, $p_{x_0}(x_0)$, are to be considered as design parameters. There are however more subtle, but still very important, design parameters. These are the importance density, and the resampling procedure. In fact different choices of these latter design parameters give rise to different names of the algorithm in the literature, as will be explained below. Having read this the reader might object by saying that this is just a matter of your definition of model, algorithm, and design parameters, which of course is true. However, when it comes to particle filter a lot can probably be gained by finding suitable definitions for model, algorithm, and design parameters. This can probably introduce more structure into all the different particle filter algorithms available today. Instead

of having a lot of different algorithms, we will just have a few algorithms, and several design parameters. The algorithms and the design parameters can then be combined in different ways in order to serve certain purposes.

To be more concrete we will now give a few examples of how different algorithms proposed in the literature can be interpreted as instances of the generic particle filter given in Algorithm 4.2. The two main aspects in which the algorithms differ are in the choice of importance density and in the resampling step.

First of all we will discuss the *sequential importance sampling/resampling (SIR)* algorithm for Model 3. This is the algorithm first introduced in the paper by Gordon *et al.* (1993), where it was referred to as the Bayesian bootstrap. The SIR algorithm can be derived from the generic particle filter 4.2 by assuming that

- The importance density is chosen according to $\pi(x_t|x_{t-1}^{(i)}, Y_t) = p(x_t|x_{t-1}^{(i)})$.
- The resampling step is applied at every time instance, i.e., we do not bother about $\hat{N}_{\text{eff}}$.

This results in the algorithm given below.

Algorithm 4.3 (Sequential importance sampling/resampling (SIR))

1. *Initialization: For* $i = 1, \ldots, N$, *initialize the particles,* $x_{0|-1}^{(i)} \sim p_{x_0}(x_0)$.
2. *For* $i = 1, \ldots, N$, *evaluate the importance weights* $q_t^{(i)} = p(y_t|x_{t|t-1}^{(i)})$ *and normalize* $\tilde{q}_t^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^{N} q_t^{(j)}}$.
3. *Measurement update: Resample with replacement* N *particles according to*
$$\Pr(x_{t|t}^{(i)} = x_{t|t-1}^{(j)}) = \tilde{q}_t^{(j)} \tag{4.38}$$
4. *Time update: For* $i = 1, \ldots, N$, *predict new particles according to*
$$x_{t+1|t}^{(i)} \sim p(x_{t+1|t}|x_{t|t}^{(i)}) \tag{4.39}$$
5. *Set* $t := t + 1$ *and iterate from step 2.*

This algorithm is very simple to implement, and in the subsequent section we provide a MATLAB implementation. Since the resampling procedure is performed at every recursion of the algorithm the only measurement update is the resampling stage in this algorithm.

One of the problems with the SIR-algorithm above is that the importance density is independent of the measurement, y_t, and hence the state-space is explored without *direct* knowledge of the observations. Here we have a window of opportunity. The performance can be improved by choosing an importance density that takes the information from the measurement into account. This leads to the *auxiliary particle filter*, first introduced in Pitt and Shephard (1999).

Another particle filtering algorithm hinted at above is the *regularized particle filter*, introduced in Musso *et al.* (2001). The idea here is to change the resampling step in order to decrease the sample impoverishment problem by resampling from a continuous kernel approximation

$$\hat{p}_N^c(x_t|Y_t) = \sum_{i=1}^{N} \tilde{q}_t^{(i)} K(x_t - x_t^{(i)}) \tag{4.40}$$

of the discrete PDF

$$\hat{p}_N(x_t|Y_t) = \sum_{i=1}^{N} \tilde{q}_t^{(i)} \delta(x_t - x_t^{(i)}) \tag{4.41}$$

In (4.40) we have used $K(\cdot)$ to denote the so called kernel density. The details will not be discussed here.

It would be very instructive to have guidelines for the user on how to choose the different design parameters in the particle filter for a certain problem. There are some guidelines of this type available in the literature, but they are scattered in various articles and hence not easily accessible to the user. For example, in Doucet (1998) there is some information on the choice of importance density and in Arulampalam *et al.* (2002) the authors stress the point that in designing a particle filter the choice of importance density is crucial.

4.3.4 Implementation

The purpose of this section is to make the particle filter more accessible to those who have still not tried it out. The algorithms have previously been given. However, there is still a gap between the algorithms and a working implementation of the particle filter. We will in this section try to fill this gap. Having read this section you will be able to implement your first particle filter from scratch within 5 minutes. We have chosen to implement the SIR particle filter given in Algorithm 4.3 and the comments in the code below are with respect to this algorithm. Before we give the implementation there are a few steps in the algorithm that are probably worth commenting. As for the resampling step the reader is referred to (Bergman, 1999) and (Ripley, 1988). Furthermore, in step 4, we want to sample $x_{t+1|t}^{(i)} \sim p(x_{t+1|t}|x_{t|t}^{(i)})$. This can be done by first generating a sample of the process noise, $w_t^{(i)} \sim p_{w_t}(w_t)$. Then we obtain $x_{t+1|t}^{(i)}$ using (2.5a), i.e.,

$$x_{t+1|t}^{(i)} = f(x_{t|t}) + w_t^{(i)}. \tag{4.42}$$

We are now ready to give the MATLAB-implementation for Algorithm 4.3 using Model 3, with Gaussian noise.

Code 1 (MATLAB-code for Algorithm 4.3 using Model 3)

```
function [xhat] = SIR(y,f,h,pe,Q,P0,N)
x = sqrt(P0)*randn(1,N);        % STEP 1, Initialize the particles
for t = 1:100
  e = repmat(y(t),1,N) - h(x);  % STEP 2, Calculate weights
  q = feval(pe,e);              % The likelihood function
  q = q/sum(q);                 % Normalize the importance weights
  ind = resampling(q);          % STEP 3, Measurement update
  x = x(:,ind);                 % The new particles
  xhat(t) = mean(x);            % Compute the estimate
  x = feval(f,x,t)+sqrt(Q)*randn(1,N); % STEP 4, Time update
end

function [i] = resampling(q)
P = cumsum(q);     N = length(q);
u = cumprod(rand(1,N).^(1./(N:-1:1)));
ut = fliplr(u);    k = 1;    i = zeros(1,N);
for j = 1:N
  while(P(k)<ut(j))
    k = k + 1;
  end;
  i(j) = k;
end;
```

The two first input arguments to the `SIR` function are the model equations, `f` and `h`, which are defined as either `inline`-objects or m-files. The other input arguments are the covariance matrices for the state, `Q`, and the measurement noise, `R`, the likelihood function, `pe`, initial state covariance, `P0`, the number of particles, `N`, and finally the measurements, `y`. The use of Code 1 is exemplified in the example provided below.

Example 4.1

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

$$x_{t+1} = \frac{x_t}{2} + \frac{25x_t}{1+x_t^2} + 8\cos(1.2t) + w_t, \tag{4.43a}$$

$$y_t = \frac{x_t^2}{20} + e_t, \tag{4.43b}$$

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 clearly a discrete-time nonlinear time-varying system with additive noise, i.e., Model 3 previously defined in Chapter 2. This system has been analyzed in many publications (see e.g., Arulampalam *et al.*, 2002; Doucet, 1998; Gordon *et al.*, 1993; Kitagawa, 1996). Hence, the result obtained from this example can be compared with the results reported in these publications.
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 SIR particle filter to obtain the results. The code for this is given below.

```
N = 1000;                    % Number of particles
P0 = 5;                      % Initial process 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) = sqrt(P0)*randn(1);   % Initial state value
y(1) = feval(h,x(1)) + sqrt(R)*randn(1);
for t = 2:100               % Simulate the system
  x(t) = feval(f,x(t-1),t) + sqrt(Q)*randn(1);
  y(t) = feval(h,x(t)) + sqrt(R)*randn(1);
end
xTrue = x;

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

Now, executing this code gives us the result shown in Figure 4.1.

We conclude this section by stressing the fact that in implementing the particle filter it is important to differ between model, algorithm, and design parameters. If this is properly done the resulting code will be well structured.

4.4 Variance Reduction by Marginalization

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 substructure available in the model equations this can be used to obtain estimates with lower

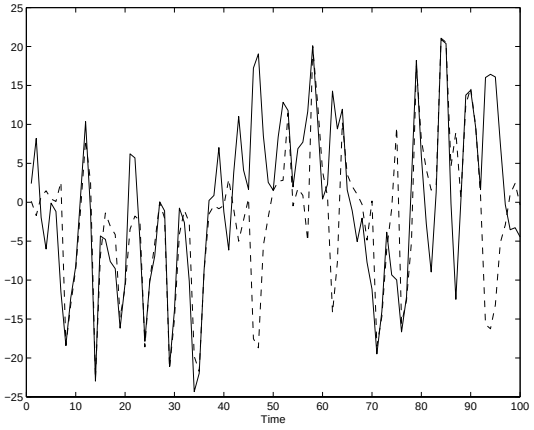


Figure 4.1 *The solid line corresponds to the true state and the dashed line stems from the estimate provided by the SIR particle filter given in Algorithm 4.3. The underlying system is given in (4.43).*

variance (Chen and Liu, 2000; Doucet, 1998; Doucet *et al.*, 1999). The reason is that the corresponding states can be optimally estimated using the Kalman filter (Anderson and Moore, 1979). Let us now assume that there indeed is a linear substructure available in the model. We can then partition the state vector according to

$$x_t = \begin{bmatrix} x_t^l \\ x_t^n \end{bmatrix}, \tag{4.44}$$

where x_t^l and x_t^n are used to denote the linear and the nonlinear state variables respectively. The basic idea is now to use Bayes' theorem according to

$$p(X_t^l, X_t^n|Y_t) = p(X_t^l|X_t^n, Y_t)p(X_t^n|Y_t). \tag{4.45}$$

The Kalman filter can be used to optimally estimate the linear probability density function $p(X_t^l|X_t^n, Y_t)$ (recall that this density is parameterized using two parameters, the mean and the covariance). The probability density function for the nonlinear state variables is estimated using the particle filter. Using the state partition (4.44) we can now write (4.1) using Bayes' theorem (4.45) according to

$$I(g(X_t)) = \frac{\int \left[\int g_t(X_t^l, X_t^n)p(Y_t|X_t^l, X_t^n)p(X_t^l|X_t^n)dX_t^l\right] p(X_t^n)dX_t^n}{\int \left[\int p(Y_t|X_t^l, X_t^n)p(X_t^l|X_t^n)dX_t^l\right] p(X_t^n)dX_t^n} \tag{4.46a}$$

$$= \frac{\int f(X_t^n)p(X_t^n)dX_t^n}{\int p(Y_t|X_t^n)p(X_t^n)dX_t^n}, \tag{4.46b}$$

where

$$f(X_t^n) \triangleq \int g_t(X_t^l, X_t^n) p(Y_t|X_t^l, X_t^n) p(X_t^l|X_t^n) dX_t^l. \tag{4.47}$$

Hence, we have analytically marginalized the linear state variables. This is the reason why this procedure of using both the Kalman filter and the particle filter will be referred to as *marginalization*. Another name commonly used in the literature is *Rao-Blackwellization*. All integrations involving the linear state variables can be evaluated on-line using the Kalman filter, Theorem 2.3. The details can be found in Paper C.

Let us call the estimated *a posteriori* PDF from the standard particle filter $\hat{p}_N^s(X_t^l, X_t^n|Y_t)$, analogously we call it $\hat{p}_N^m(X_t^l, X_t^n|Y_t)$ when the marginalized particle filter has been used. They can be written according to

$$\hat{p}_N^s(X_t^l, X_t^n|Y_t) = \sum_{i=1}^{N} \tilde{q}_t(X_t^{n,(i)}, X_t^{l,(i)}) \delta(X_t - X_t^{(i)}), \tag{4.48a}$$

$$\hat{p}_N^m(X_t^n|Y_t) = \sum_{i=1}^{N} \tilde{q}(X_t^{n,(i)}) \delta(X_t^n - X_t^{n,(i)}). \tag{4.48b}$$

If we use the standard particle filter for all states the dimension of the space in which the particles live will be $n_{x_t} = \dim x_t$, whereas if we use the marginalized particle filter the corresponding dimension will be $n_{x_t^n} = \dim x_t^n$. Intuitively, since $\dim x_t^n < \dim x_t$ we have to use more particles to approximate $\hat{p}_N^s(X_t^l, X_t^n|Y_t)$, than to approximate $\hat{p}_N^m(X_t^l, X_t^n|Y_t)$, in order to reach the same accuracy in the approximations. This is formally proved in Doucet *et al.* (1999), by showing that the variance of the estimates provided from $\hat{p}_N^s(X_t^l, X_t^n|Y_t)$ are larger than or equal to the variance of the estimates from $\hat{p}_N^m(X_t^l, X_t^n|Y_t)$.

The idea of using a filter combined of a Kalman filter for the linear state variables and a particle filter for the nonlinear state variables is certainly not a new one. It has previously been used in the literature (see e.g., Chen and Liu, 2000; Doucet, 1998; Doucet *et al.*, 2001b; Nordlund, 2002). Our contribution is that we in detail explain and sort out the details on how to use the marginalized particle filter for the quite general nonlinear state-space model,

$$x_{t+1}^n = f_t^n(x_t^n) + A_t^n(x_t^n) x_t^l + G_t^n(x_t^n) w_t^n, \tag{4.49a}$$

$$x_{t+1}^l = f_t^l(x_t^n) + A_t^l(x_t^n) x_t^l + G_t^l(x_t^n) w_t^l, \tag{4.49b}$$

$$y_t = h_t(x_t^n) + C_t(x_t^n) x_t^l + e_t, \tag{4.49c}$$

In Schön *et al.* (2003c), Paper C in this thesis, the results and the complete derivations can be found. We will here illustrate the idea by providing the schematic algorithm below.

Algorithm 4.4 (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. *Evaluate the importance weights and normalize.*
3. *Particle filter measurement update: 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. *Set* $t := t + 1$ *and iterate from step 2.*

The only difference from the SIR particle filter presented in detail in Algorithm 4.3 is in step 4, where we now have introduced two additional steps. These two steps correspond to the optimal estimation of the linear state variables using the Kalman filter. For a thorough discussion concerning the details the reader is referred to Paper C.

4.5 Applications

Since the paper by Gordon *et al.* (1993) a large number of publications of the particle filter have appeared. The current state of the art is summarized in Doucet *et al.* (2001a). During the five years following the paper mentioned above the theory was mainly developed by statisticians. When the theory was mature enough the signal processing community quickly adopted the results, and developed practical algorithms for applications.

Given this increasing number of applications of the Bayesian ideas such as the particle filter it is perhaps time to answer the question Peterka posed in Peterka (1979)

> Lindley (1975) in his talk on the future of statistics forecasts, following Finetti (1974), that the full change in statistics towards Bayesian ideas will come round 2020. Can we engineers afford to wait that long?

The answer is; No, we engineers cannot afford to wait that long. Practical applications of the particle filter include, just to mention a few, target tracking (Gordon *et al.*, 1993; Karlsson, 2002), navigation (Bergman, 1999; Frykman, 2003; Karlsson, 2003; Nordlund, 2002), audio source separation (Andrieu and Godsill, 2000), underwater navigation (Karlsson, 2003), mobile robot localization (Fox *et al.*, 2001; Jensfelt, 2001; Wijk, 2001), positioning (Gustafsson *et al.*, 2002; Hall, 2000). The company NIRA Dynamics in Linköping have seen the commercial value of the particle filter, and use it in a positioning and navigation system that is currently under development.

4.5.1 System Identification

Within the field of signal processing the particle filter has gained a lot of attention. However, so far the impact on the automatic control and the system identification societies has been quite limited. In Paper D we try to show that the particle filter offers a general tool for estimating unknown parameters in nonlinear models of moderate complexity. In this section we will give the overall ideas, for the details the reader is referred to Paper D, (Schön and Gustafsson, 2003).

In Section 2.4 it was explained that the foundation for using the particle filter in a system identification context was given in Ljung and Söderström (1983) where they posed the system identification problem as a Bayesian state estimation problem, simply by augmenting the state vector, x_t, with the parameter vector, θ, according to

$$\bar{x}_t = \begin{bmatrix} x_t \\ \theta \end{bmatrix}. \tag{4.50}$$

The idea is then to consider the nonlinear state estimation problem w.r.t. this extended state. The corresponding model is given by

$$\begin{bmatrix} x_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} f_t(x_t, \theta_t) \\ \theta_t \end{bmatrix} + \begin{bmatrix} w_t^x \\ w_t^\theta \end{bmatrix} \tag{4.51a}$$

$$y_t = h_t(x_t, \theta_t) + e_t, \tag{4.51b}$$

which falls in the class of models defined by Model 3 in Section 2.2.1. Hence, we are back in a formulation where the particle filter framework can be readily applied.

One of the fundamental problems with applying the particle filter to the problem of system identification is that parameters are assumed to be time-invariant. Whereas the particle filter is designed to estimate dynamic entities. We have proposed to tackle this problem by modeling the parameter as a random walk

$$\theta_{t+1} = \theta_t + w_t, \tag{4.52}$$

where the variance of the noise, w_t, tends to zero as the estimate of the parameter converges. This is sort of the same idea used in the recursive least squares algorithm or the Kalman filter for parameter estimation, where the step size initially decays as $1/t$ and then converges/fluctuates around its constant value. Another interesting approach to handling this problem is taken in Andrieu and Godsill (2000) using Markov chain Monte Carlo (MCMC) methods. The ideas presented in Andrieu *et al.* (2001) are interesting, since they consider the use of marginalization in bilinear system, where the marginalized particle filter is used to estimate both parameters and states.

In Paper D we give the details on how to apply the particle filter to the nonlinear system identification problem. We also show how to improve the estimates using the marginalization technique introduced in Section 4.4. The ideas have successfully been applied to a nonlinear chaotic system.

5

Concluding Remarks

In this first part we have introduced the most important background theory for the research presented in this thesis. We have also established connections between our research and the presently available theory, by hinting at our results. However, for the details the reader has been referred to Part II. Below we will give a brief summary of the results, as well as hinting at interesting topics for future research.

A very important tool for tackling the problem of nonlinear estimation has in this thesis been the sequential Monte Carlo methods, i.e., the particle filters, which were introduced in Chapter 4. In Section 4.5.1 a very brief introduction to using particle filter for the system identification problem was given. The parameters were augmented to the state vector and the resulting problem was handled using the standard and the marginalized particle filter. The details are given in Paper D. A very interesting topic for further research is to investigate the relationship between using roughening noise as we did in Paper D and using MCMC methods instead as was done in Andrieu and Godsill (2000). Another interesting issue is to sort out the computational complexity issues involved in using the marginalized particle filter introduced in Section 4.4 and thoroughly discussed in Paper C. This is especially interesting when we consider bilinear systems as was also noted in Andrieu and Godsill (2000). There are probably several problems that can gain from using the marginalized particle filters. There are two main reasons for this. First of all the dimension of the space in which the particles live is reduced as compared to the standard particle filter. Secondly, the marginalized particle filter will provide

estimates with better or the same quality as the standard particle filter.

In Section 2.2 different model classes were discussed. The models resulting from the new object-oriented software tools are constituted of differential-algebraic equations. The standard model class used today is the state-space model, in other words ordinary differential equations. Since the DAE model class is larger than the class described by state-space descriptions there is a need to extend the theory of estimation to handle DAE models as well. In Section 2.2.2 we explain how to estimate the internal variables in linear differential-algebraic equations. We also show that the main thing when it comes to this problem is to solve the problem of how to incorporate white noise in this type of equations. A detailed treatment of this problem is given in Paper A. It would indeed be very interesting to extend these results to the nonlinear case. If this was possible the particle filter could be used for estimating the internal variables.

In Chapter 3 we discussed deterministic estimation and in Section 3.3 we explained how to pose the estimation problem as a convex optimization problem. A very appealing property of this formulation is that it is straightforward to include linear equality and convex inequality constraints to this formulation and the problem can still be solved using standard software. These constraints typically correspond to prior information on the problem at hand. In Paper B these ideas were discussed in more detail and an example was given, showing how to use prior information in the problem of estimation. In Problem 4 we defined the moving horizon estimation problem. This formulation can probably be used for nonlinear change detection. One of the main drawbacks with the constrained estimation problems are that the solution requires a lot of computational resources. Hence, it would be very useful to find a recursive formulation of the solution to constrained convex optimization problems.

Bibliography

Alspach, D. and Sorenson, H. (1972). Nonlinear Bayesian estimation using Gaussian sum approximation. *IEEE Transactions on Automatic Control*, 17:439–448.

Anderson, B. and Moore, J. (1979). *Optimal Filtering*. Information and system science series. Prentice Hall, Englewood Cliffs, New Jersey.

Andersson, S. (2002). *Hidden Markov Models - Traffic Modeling and Subspace Methods*. PhD thesis, Centrum for Mathematical Sciences, Mathematical Statistics, Lund University, Lund, Sweden.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. (2003). An introduction to mcmc for machine learning. *Machine Learning*, 50:5–43.

Andrieu, C., Doucet, A., and Fitzgerald, W. (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. and Godsill, S. (2000). A particle filter for model based audio source separation. In *International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000)*, Helsinki Finland.

Arulampalam, M., 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.

Ascher, U. and Petzold, L. (1998). *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia.

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*. PhD thesis, Linköping University. Dissertation No. 579.

Brenan, K., Campbell, S., and Petzold, L. (1996). *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM's Classics in Applied Methematics. SIAM, New York.

Bucy, R. and Senne, K. (1971). Digital synthesis on nonlinear filters. *Automatica*, pages 287–298.

Campbell, S. (1990). Descriptor systems in the 90's. In *proceedings of the 29th Conference on Decision and control*, pages 442–447, Honolulu, Hawaii, USA. IEEE.

Chen, R. and Liu, J. (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society*, 62(3):493–508.

Cox, H. (1964). On the estimation of state variables and parameters for noisy dynamic systems. *IEEE Transactions on Automatic Control*, 9:5–12.

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 *Preprints of the 10th IFAC World Congress, Munich, Germany*, volume 9, pages 211–215.

Dai, L. (1989). Filtering and LQG problems for discrete-time stochastic singular systems. *IEEE Transactions on Automatic Control*, 34(10):1105–1108.

Darouach, M., Zasadzinski, M., and Mehdi, D. (1993). State estimation of stochastic singular linear systems. *International Journal of Systems Sciences*, 24(2):345–354.

Deng, Z. and Liu, Y. (1999). Descriptor Kalman filtering. *International Journal of Systems Science*, 30(11):1205–1212.

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.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.

Doucet, A., Gordon, N., and Krishnamurthy, V. (1999). Particle filters for state estimation of jump Markov linear systems. Technical Report CUED/F-INFENG/TR 359, Signal Processing Group, Department of Engineering, University of Cambridge, Trupington street, CB2 1PZ Cambridge.

Doucet, A., Gordon, N., and Krishnamurthy, V. (2001b). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624.

Eckhardt, R. (1987). Stan Ulam, John von Neumann, and the Monte Carlo method. *Los Alamos Science*, pages 131–137.

Elliott, R., Aggoun, L., and Moore, J. (1995). *Hidden Markov Models Estimation and Control*. Springer.

Fearnhead, P. (1998). *Sequential Monte Carlo methods in filter theory*. PhD thesis, University of Oxford.

Finetti, B. D. (1974). *Theory of Probability: A Critical Introductory Treatment, Vol. 1*. Wiley, New York.

Fisher, R. (1912). On an absolute criterion for fitting frequency curves. *Messenger of Mathematics*, 41:155–160.

Fox, D., Thrun, S., Burgard, W., and Dellaert, F. (2001). Chapter 19. particle filters for mobile robot localization. In Doucet, A., de Freitas, N., and Gordon, N., editors, *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, pages 401–428. Springer.

Frykman, P. (2003). Applied particle filters in integrated aircraft navigation. LiTH-ISY-EX-3406, Automatic control and communications systems, Linköping university.

Gantmacher, F. (1959). *The Theory of Matrices*, volume 2. Chelsea publishing company, New York.

Gardner, W. (1986). *Introduction to Random Processes with Applications to Signals and Systems*. Macmillan publishing company, New York.

Gerdin, M., Glad, T., and Ljung, L. (2003). Parameter estimation in linear differential-algebraic equations. In *proceedings of the 13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands. IFAC.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339.

Ghaoui, L. E. and Lebret, H. (1997). Robust solutions to least-squares problems with uncertain data. *SIAM Journal on matrix analysis and applications*, 18(4):1035–1064.

Goodwin, G. (2003). Inverse problems with constraints. In *proceedings of the 15th Triennial World Congress*, Barcelona, Spain. IFAC.

Gordon, N., Salmond, D., and Smith, A. (1993). A 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.

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., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J. (2002). Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437.

Hall, P. (2000). A Bayesian approach to map-aided vehicle positioning. Master's thesis, Department of Electrical Engineering, Linköping University.

Handschin, J. (1970). Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica*, 6:555–563.

Handschin, J. and Mayne, D. (1969). Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control*, 9:547–559.

Hastings, W. (1970). Monte Carlo simulation methods using Markov Chains and their applications. *Biometrica*, 57:97–109.

Ho, Y. and Lee, R. (1964). A Bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, 9:333–338.

Jazwinski, A. (1970). *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York.

Jensfelt, P. (2001). *Approaches to Mobile Robot Localization in Indoor Environments*. PhD thesis, Department of Signals, Royal Institute of Technology.

Kailath, T. (1980). *Linear systems*. Prentice Hall, Englewood Cliffs, N.J.

Kailath, T., Sayed, A., and Hassibi, B. (2000). *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, New Jersey.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Trans. AMSE, J. Basic Engineering*, 82:35–45.

Karlsson, R. (2002). Simulation based methods for target tracking. Licenciate thesis, Linköping university. Thesis No. 930.

Karlsson, R. (2003). Particle filter and Cramér-Rao lower bound for underwater navigation. In *proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong.

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.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of American Statistical Association*, 89(425):278–288.

Kramer, S. and Sorenson, H. (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.

Lindley, D. (1975). The future of statistics - a Bayesian 21st century. In *proceedings Conference on Directions for Mathematical Statistics*, pages 106–115. Supplement to Adv. Appl. Probab.

Ljung, L. (1999). *System identification, Theory for the user*. System sciences series. Prentice Hall, Upper Saddle River, NJ, 2nd edition.

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.

Lobo, M., Vandenberghe, L., Boyd, S., and Lebret, H. (1998). Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228.

Löfberg, J. (2003). *YALMIP Yet another LMI parser*. Linköping university. available at: www.control.isy.liu.se/ johanl/yalmip.html.

Maciejowski, J. (2002). *Predictive Control with Constraints*. Prentice Hall.

Mattsson, S., Elmqvist, H., and Otter, M. (1998). Physical system modeling with modelica. *Control Engineering Practice*, 6:501–510.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., 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. (1995). Moving horizon observers and observer-based control. *IEEE Transactions on Automatic Control*, 40(6):995–1006.

Musso, C., Oudjane, N., and Gland, F. L. (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.

Nikoukhah, R., Campbell, S., 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, Florida, USA.

Nikoukhah, R., Campbell, S., and Delebecque, F. (1999). Kalman filtering for general discrete-time linear systems. *IEEE Transactions on Automatic Control*, 44(10):1829–1839.

Ninness, B. and Henriksen, S. (2003). System identification via a computational Bayesian approach. In *proceedings of the 13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands.

Nordlund, P.-J. (2002). Sequential Monte Carlo filters and integrated navigation. Licenciate thesis, Linköping university. Thesis No. 945.

Øksendal, B. (2000). *Stochastic Differential Equations An Introduction with Applications*. Springer, 5 edition.

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. Pergamon Press.

Pitt, M. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599.

Rao, C. (2000). *Moving Horizon Strategies for the Constrained Monitoring and Control of Nonlinear Discrete-Time Systems*. PhD thesis, University of Wisconsin Madison.

Rao, C., Rawlings, J., and Lee, J. (2001). Constrained linear state estimation - a moving horizon approach. *Automatica*, pages 1619–1628.

Ripley, B. (1988). *Stochastic Simulation*. John Wiley.

Robert, C. and Casella, G. (1999). *Monte Carlo Statistical Methods*. Springer texts in statistics. Springer, New York.

Robertson, D. and Lee, J. (2002). On the use of constraints in least squares estimation and control. *Automatica*, 38:1113–1123.

Rugh, W. (1996). *Linear System Theory*. Information and system sciences series. Prentice Hall, Upper Saddle River, New Jersey, 2 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.

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. Accepted for publication.

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.

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. (2003c). Marginalized particle filters for nonlinear state-space models. *Submitted to IEEE Transactions on Signal Processing*.

Smith, A. and Gelfand, A. (1992). Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician*, 46(2):84–88.

Sorenson, H. (1970). Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum*, 7:63–68.

Sorenson, H. (1974). On the development of practical nonlinear filters. *Information Sciences*, 7:253–270.

Sorenson, H. (1988). Chapter 6. Recursive estimation for nonliear dynamic systems. In Spall, J., editor, *Bayesian Analysis of Time Series and Dynamic Models*, pages 127–165. Marcel Dekker inc., New York.

Sorenson, H. and Alspach, D. (1971). Recursive Bayesian estimation using Gaussian sum. *Automatica*, 7:465–479.

Svenzén, N. (2002). Real time implementation of map aided positioning using a Bayesian approach. LiTH-ISY-EX-3297-2002, Automatic control and communications systems, Linköping university.

Tiller, M. (2001). *Introduction to Physical Modeling with Modelica*. Kluwer Academic Press, Boston.

Vandenberghe, L. and Boyd, S. (2001). Convex optimization. To be published.

Varga, A. (1992). Numerical algorithms and software tools for analysis and modelling of descriptor systems. In *proceedings of 2nd IFAC Workshop on System Structure and Control, Prague, Czechoslovakia*, pages 392–395.

Wijk, O. (2001). *Triangulation Based Fusion of Sonar Data with Application in Mobile Robot Mapping and Localization*. PhD thesis, Department of Signals, Royal Institute of Technology, Stockholm.

Winkler, R. (2003). Stochastic differential algebraic equations of index 1 and applications in circuit simulation. *Journal of Computational and Applied Mathematics*, pages 477–505.

Zhu, Y. and Li, X. (1999). Recursive least squares with linear constraints. In *proceedings of the 38th Conference on Decision & Control*, pages 2414–2419, Phoenix, Arizona, USA. IEEE.

Part II

Publications

A

A Modeling and Filtering Framework for Linear Differential-algebraic Equations

Authors: Thomas Schön, Markus Gerdin, Torkel Glad and Fredrik Gustafsson.

Edited version of paper accepted for publication in *proceedings of the 42th IEEE Conference on Decision and Control*, Hawaii, USA, 2003.

Preliminary version published as Technical Report LiTH-ISY-R-2498, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

A Modeling and Filtering Framework for Linear Differential-algebraic Equations

Thomas Schön, Markus Gerdin, Torkel Glad and Fredrik Gustafsson.

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

Abstract

General approaches to modeling, for instance using object-oriented software, lead to differential-algebraic equations (DAE). As the name reveals, it is a combination of differential and algebraic equations. For state estimation using observed system inputs and outputs in a stochastic framework similar to Kalman filtering, we need to augment the DAE with stochastic disturbances ("process noise"), whose covariance matrix becomes the tuning parameter. We will determine the subspace of possible disturbances based on the linear DAE model. This subspace determines all degrees of freedom in the filter design, and a Kalman filter algorithm is given. We illustrate the design on a system with two interconnected rotating masses.

Keywords: Differential-algebraic equations, Implicit systems, Descriptor systems, Singular systems, White noise, Noise, Discretization, Kalman filter.

1 Introduction

In recent years so-called object-oriented modeling software has increased in popularity. Examples of such software are Omola, Dymola, the SimMechanics toolbox for MATLAB, and Modelica (Mattsson *et al.*, 1998; Tiller, 2001). Such modeling software makes it possible to model physical systems by connecting sub-models in a way which parallels the physical construction and without having to manually manipulate any equations. The available software usually gives the user the possibility to simulate the system, and perhaps also to extract a structured model in

an automatic way. This model generally becomes a differential-algebraic equation (DAE), which in the linear case can be written

$$E\dot{x}(t) + Fx(t) = B_u u(t), \tag{1a}$$

where $x(t)$ is the internal variable vector, $u(t)$ is the system input vector and E, F, B_u are matrices of appropriate dimensions. We assume that E is singular, otherwise we get an ordinary differential equation (ODE) by simply multiplying with E^{-1} from the left, and the standard Kalman filtering theory applies. Hence, when E is singular we obtain a differential-algebraic equation and the reason for the singularity is often that purely algebraic equations are present. Other common names for the model structure (1a) are e.g., implicit systems, descriptor systems, semi-state systems, generalized systems, and differential equations on a manifold (Campbell, 1990).

We have the possibility to place sensors in the system to get a measurement equation

$$y(t) = Cx(t) + e(t), \tag{1b}$$

where $y(t)$ is the measurement and $e(t)$ the sensor noise. An important special case we will discuss separately is for computer controlled systems, where the measurements $y[k]$ are available at the sampling times $t = kT_s$,

$$E\dot{x}(t) + Fx(t) = B_u u(t), \tag{2a}$$
$$y[kT_s] = Cx(kT_s) + e[kT_s]. \tag{2b}$$

The estimation problem is to estimate $x(t)$ from $y[kT_s]$. There are two reasons why we have to introduce process noise to (2a):

- There are unmodeled dynamics and disturbances acting on the system, that can only be included in the model as an unknown stochastic term.
- There is a practical need for tuning the filter in order to make a trade-off between tracking ability and sensor noise attenuation. This is in the Kalman filter accomplished by keeping the sensor noise covariance matrix constant and tuning the process noise covariance matrix, or the other way around. Often, it is easier to describe the sensor noise in a stochastic setting, and then it is more natural to tune the process noise.

With process noise, the model (1) becomes

$$E\dot{x}(t) + Fx(t) = B_u u(t) + B_w w(t), \tag{3a}$$
$$y(t) = Cx(t) + e(t). \tag{3b}$$

The problem is to determine where in the system disturbances can occur. To fit the optimal filtering and Kalman filtering framework, $w(t)$ should be white noise. As will be demonstrated, adding white noise to all equations can lead to derivatives of

white noise affecting internal variables of the system directly. This will be referred to as a noncausal system, with a physical interpretation of infinite forces, currents etc. Therefore, we will derive a basis for the subspace of all possible disturbances, that leads to causal systems. This basis is taken as B_w in (3), and the process noise covariance matrix $Q = \mathrm{Cov}[w(t)]$ is used as the design variable to rotate and scale this basis. This is a new way of defining the process noise as far as we know. The problem itself, however, is addressed in Campbell (1990), where it is suggested to use band limited noise to avoid these problems. The idea is that the derivative of such noise exists, but the drawback is that the Kalman filter will become sub-optimal.

A system with the same structure as (3) but in discrete time will be referred to as a discrete-time descriptor system. Such systems may also be noncausal, but are easier to handle since the noncausality here means dependence on future values of the noise or the input. An application for such systems is discrete-time state-space systems with constraints. For an example see Schön *et al.* (2003). In the discrete time case much work has already been done (for example on Kalman filtering see e.g., Dai, 1987, 1989a; Darouach *et al.*, 1993; Deng and Liu, 1999; Nikoukhah *et al.*, 1998, 1999). In the continuous time case much less work has been done on statistical methods. However, some attempts to introduce white noise in the continuous case has been done as well (see e.g., Schein and Denk, 1998; Winkler, 2003).

2 Derivation of the Process Noise Subspace

We will omit the deterministic input in this derivation for notational convenience, so the continuous-time linear invariant differential-algebraic equations considered has the form (4). The reader is referred to Gerdin *et al.* (2003) for details on how the noncausality with respect to the input signal, $u(t)$, can be handled.

$$E\dot{x}(t) + Fx(t) = Bw(t) \tag{4a}$$

$$y(t) = Cx(t) + e(t) \tag{4b}$$

The E, F, and C matrices in (4) are constant matrices. For the purpose of this discussion we will assume that w and e are continuous-time white noises. (See (Åström, 1970) for a thorough treatment of continuous-time white noise). If $\det[Es + F]$ is not identically zero as a function of $s \in \mathbb{R}$, (4) can always be transformed into the *standard form* (6) (Brenan *et al.*, 1996). Note that if $\det[Es + F]$ is identically zero, then $x(t)$ is not uniquely determined by $w(t)$ and the initial value $x(0)$. This can be realized by Laplace transforming (4). Therefore it is a reasonable assumption that $\det[Es + F]$ is not identically zero.

2.1 Time-domain Derivation

First, a transformation to the standard form is needed. This is done by finding a suitable change of variables $x = Qz$ and a matrix P to multiply (4a) from the left.

Both P and Q are nonsingular matrices. By doing this we get

$$PEQ\dot{z}(t) + PFQz(t) = PBw(t), \tag{5}$$

which for suitably chosen P- and Q-matrices can be written in the following standard form:

$$\begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} + \begin{bmatrix} -A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} w(t), \tag{6}$$

where the N-matrix is *nilpotent*, i.e., $N^k = 0$ for some k. The matrices P and Q can be calculated using, e.g., ideas from Varga (1992) involving the generalized real Schur form and the generalized Sylvester equation. We can also write (6) on the form (7) (Dai, 1989b; Ljung and Glad, 2003).

$$\dot{z}_1(t) = Az_1(t) + G_1 w(t), \tag{7a}$$

$$z_2(t) = \sum_{i=0}^{k-1} (-N)^i G_2 \frac{d^i w(t)}{dt^i}. \tag{7b}$$

From a theoretical point of view G_1 can be chosen arbitrarily, since it describes how white noise should enter an ordinary differential equation. However, constraints on G_1 can of course be imposed by the physics of the system that is modeled. When it comes to G_2, the situation is different, here we have to find a suitable parameterization. The problem is now that white noise cannot be differentiated, so we proceed to find a condition on the B-matrix in (4a) under which there does not occur any derivatives in (7b), i.e., $N^i G_2 = 0$ for all $i \geq 1$. This is equivalent to that $NG_2 = 0$. The result is given in the following theorem.

Theorem A.1
The condition to avoid to differentiate white noise is equivalent to requiring that

$$B \in \mathcal{R}(M), \tag{8}$$

where M is a matrix derived from the standard form (6) (see the proof for details on how M is derived).

The expression $B \in \mathcal{R}(M)$ means that B is in the *range* of M, that is the columns of B are linear combinations of the columns of M.

Proof Let the $n \times n$ matrix N in (6) have the singular value decomposition (SVD)

$$N = UDV^T. \tag{9}$$

Since it is nilpotent it is also singular, so m diagonal elements in D are zero. Partition $V = [V_1, V_2]$, where V_2 contains the last m columns of V having zero

singular values. Then $NV_2 = 0$, and we can write $G_2 = V_2T$, where T is an arbitrary $m \times m$ matrix, which parameterizes all matrices G_2 that satisfies $NG_2 = 0$.

According to (5) and (6) we have

$$B = P^{-1} \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}. \tag{10}$$

If we now let $P^{-1} = [R_1,\, R_2]$, we can write (10) as

$$B = P^{-1} \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} G_1 \\ V_2T \end{bmatrix} = \underbrace{[R_1\ R_2V_2]}_{M} \begin{bmatrix} G_1 \\ T \end{bmatrix} \tag{11}$$

where both G_1 and T can be chosen arbitrarily. This calculation gives that

$$B \in \mathcal{R}(M) \tag{12}$$

is a condition for avoiding differentiation of the white noise signal $w(t)$. □

The B-matrices satisfying (12) will thus allow us to incorporate white noise without having a problem with differentiation of white noise. The design parameters to be specified are G_1 and T, defined in the proof above. Also note that the requirement that white noise should not be differentiated is related to the concept of *impulse controllability* discussed in Dai (1989b).

2.2 Frequency-domain Derivation

The same condition on the noise can be derived in the frequency domain, as shown in this section. Throughout the section, we need some concepts from the theory of matrix fraction descriptions (MFD). We start by defining the *row degree* of a polynomial matrix and the concept of a *row reduced* MFD according to Rugh (1996).

Definition A.1
The i^{th} row degree *of a polynomial matrix $P(s)$, written as $r_i[P]$, is the degree of the highest degree polynomial in the i^{th} row of $P(s)$.*

Definition A.2
If the polynomial matrix $P(s)$ is square and nonsingular, then it is called row reduced *if*

$$\deg[\det[P(s)]] = r_1[P] + \cdots + r_n[P]. \tag{13}$$

We will use the following theorem from Kailath (1980):

Theorem A.2
If $D(s)$ is row reduced, then $D^{-1}(s)N(s)$ is proper if and only if each row of $N(s)$ has degree less than or equal the degree of the corresponding row of $D(s)$, i.e., $r_i[N] \leq r_i[D], i = 1, \ldots, n$.

To utilize the results we need to write (4a) as a matrix fraction description. A MFD of (4a) is

$$X(s) = (Es + F)^{-1}BW(s). \tag{14}$$

According to Rugh (1996) a MFD can be converted to row reduced form by pre-multiplication of a unimodular[1] matrix $U(s)$. That is, $D(s)$ is row reduced in the MFD

$$X(s) = D^{-1}(s)N(s)W(s) \tag{15}$$

where $D(s) = U(s)(Es + F)$ and $N(s) = U(s)B$ for a certain unimodular matrix $U(s)$. Now, Theorem A.2 shows that the transfer function of the system is proper if the highest degree of the polynomials in each row in $N(s)$ is lower than or equal to the highest degree of the polynomials in the corresponding row of $D(s)$. This gives a condition on B in the following way:

Writing $U(s)$ as

$$U(s) = \sum_{i=0}^{m} U_i s^i \tag{16}$$

and writing the j^{th} row of U_i as U_{ij}, shows that the condition

$$U_{ij}B = 0 \quad i > r_j[D],\ j = 1 \dots n \tag{17}$$

guarantees that the transfer function of the system is proper.

Conversely, assume that (17) does not hold. Then some row degree of N(s) is higher than the corresponding row degree of D(s), so the transfer function is then according to Theorem A.2 not proper.

This discussion proves the following theorem.

Theorem A.3
The transfer function of the system (4) is proper if and only if

$$U_{ij}B = 0 \quad i > r_j[D],\ j = 1 \dots n. \tag{18}$$

Note that the criterion discussed in this section requires that the MFD is transformed to row reduced form, and an algorithm for finding this transformation is provided in Rugh (1996).

We have now proved two theorems, one using time-domain methods and one using frequency-domain methods, that gives conditions which are equivalent to that no white noise is differentiated in (4). This means that these two conditions are equivalent as well. The frequency-domain method is good in the sense that we do not have to compute the standard form (6). However, if we want to discretize the equations it is worthwhile to compute the standard form. Once this is done the celebrated Kalman filter can be used to estimate the internal variables, $x(t)$. In the subsequent section we will discuss the discretization and the estimation problems.

[1] A polynomial matrix is called unimodular if its determinant is a nonzero real number (Kailath, 1980).

3 Filtering

3.1 Discretization

If the noise enters the system according to a B-matrix satisfying Theorem A.1 or A.3 the original system (4) can be written as

$$\dot{z}_1(t) = Az_1(t) + G_1 w(t), \tag{19a}$$
$$z_2(t) = G_2 w(t), \tag{19b}$$
$$y(t) = CQz(t) + e(t). \tag{19c}$$

where $x = Qz$. Furthermore $w(t)$ and $e(t)$ are both assumed to be Gaussian white noise signals with covariances R_1 and R_2 respectively, and zero cross-covariance (the case of nonzero cross-covariance can be handled as well, the only difference is that the expressions are more involved).

System (19) can be discretized using standard techniques from linear systems theory (Rugh, 1996). If we assume that $w(t)$ remains constant during one sample interval[2], we have (here it is assumed that sampling interval is one to simplify the notation)

$$w(t) = w[k], \qquad k \le t < (k+1) \tag{20}$$

we obtain

$$z_1[k+1] = \tilde{A} z_1[k] + \tilde{G}_1 w[k], \tag{21a}$$
$$z_2[k] = G_2 w[k], \tag{21b}$$
$$y[k] = CQz[k] + e[k] \tag{21c}$$

where

$$\tilde{A} = e^A \qquad \tilde{G}_1 = \int_0^1 e^{A\tau} d\tau G_1. \tag{22}$$

Hence (21) and (22) constitutes a discrete-time model of (4).

3.2 Kalman Filter

In order to apply the Kalman filter to the discrete model (21) we start out by rewriting (21c) as

$$\begin{aligned} y[k] &= CQz[k] + e[k] = [\tilde{C}_1 \tilde{C}_2] \begin{bmatrix} z_1[k] \\ z_2[k] \end{bmatrix} + e[k] = \tilde{C}_1 z_1[k] + \tilde{C}_2 z_2[k] + e[k] \\ &= \tilde{C}_1 z_1[k] + \underbrace{\tilde{C}_2 G_2 w[k] + e[k]}_{\tilde{e}[k]} \end{aligned} \tag{23}$$

[2]See e.g., Gustafsson (2000) for a discussion on other possible assumptions on the stochastic process $w(t)$ when it comes to discretization.

Combining (21a) and (23) we obtain

$$z_1[k+1] = \tilde{A} z_1[k] + \tilde{G}_1 w[k] \tag{24a}$$

$$y[k] = \tilde{C}_1 z_1[k] + \tilde{e}[k] \tag{24b}$$

Note that the measurement noise, $\tilde{e}[k]$, and the process noise, $w[k]$, are correlated. Now, the Kalman filter can be applied to (24) in order to estimate the internal variables $z_1[k]$ and the process noise $w[k]$. Finally an estimate of the internal variables $z_2[k]$ can be found using the estimated process noise, since $z_2[k] = G_2 w[k]$, according to (21b). Finally the internal variables, $x[k]$, are obtained by $x[k] = Q^{-1} z[k]$. For details on the Kalman filter see Glad and Ljung (2000).

4 Example

In this example we will treat a system composed of two rotating masses as shown in Figure 1. The two rotating parts are described by the torques M_1, M_2, M_3 and

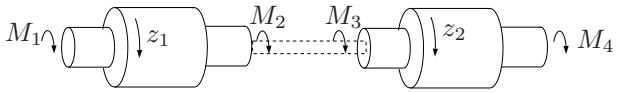


Figure 1 *Two interconnected rotating masses.*

M_4 and the angular velocities z_1 and z_2. The equations describing this system are

$$J_1 \dot{z}_1 = M_1 + M_2 \tag{25a}$$

$$J_2 \dot{z}_2 = M_3 + M_4 \tag{25b}$$

$$M_2 = -M_3 \tag{25c}$$

$$z_1 = z_2. \tag{25d}$$

Written on the form (4) these equations are

$$\begin{bmatrix} J_1 & 0 & 0 & 0 \\ 0 & J_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \dot{x} + \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \end{bmatrix} x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} M_1 \\ M_4 \end{bmatrix}, \tag{26}$$

where $x = [z_1,\, z_2,\, M_2,\, M_3]^T$. Note that the matrix in front of $\dot{x}$ is singular, hence (26) is a differential-algebraic equation. Using the following transformation matrices P and Q

$$P = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ \frac{J_2}{J_1+J_2} & -\frac{J_1}{J_1+J_2} & \frac{J_2}{J_1+J_2} & 0 \end{bmatrix}, \qquad Q = \begin{bmatrix} \frac{1}{J_1+J_2} & \frac{J_2}{J_1+J_2} & 0 & 0 \\ \frac{1}{J_1+J_2} & -\frac{J_1}{J_1+J_2} & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{27}$$

the equations can be written in the standard form (6):

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{J_1 J_2}{J_1+J_2} & 0 & 0 \end{bmatrix} \dot{z} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} z = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ \frac{J_2}{J_1+J_2} & -\frac{J_1}{J_1+J_2} \end{bmatrix} \begin{bmatrix} M_1 \\ M_4 \end{bmatrix} \tag{28}$$

Now to the important part, if we want to incorporate noise into the differential-algebraic equation (26), by adding Bw to (26), which B-matrices are allowed?

To answer this question Theorem A.1 can be consulted. We begin by calculating the matrices R_1, R_2 and V_2 from (27) and (28). We have that

$$N = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{J_1 J_2}{J_1+J_2} & 0 & 0 \end{bmatrix} \quad \Rightarrow \quad V_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{29}$$

and

$$P^{-1} = \begin{bmatrix} \frac{J_1}{J_1+J_2} & 0 & -1 & 1 \\ \frac{J_2}{J_1+J_2} & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \Rightarrow \quad R_1 = \begin{bmatrix} \frac{J_1}{J_1+J_2} \\ \frac{J_2}{J_1+J_2} \\ 0 \\ 0 \end{bmatrix}, \quad R_2 = \begin{bmatrix} 0 & -1 & 1 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{30}$$

We can now calculate the M matrix:

$$M = \begin{bmatrix} R_1 & R_2 V_2 \end{bmatrix} = \begin{bmatrix} \frac{J_1}{J_1+J_2} & -1 & 1 \\ \frac{J_2}{J_1+J_2} & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{31}$$

As the requirement was that $B \in \mathcal{R}(M)$ this simply means that we cannot directly add white noise to (25d) (if $J_1 > 0$ and $J_2 > 0$). This result makes physical sense, as a step change in the angular velocity would require an infinite torque.

The same condition on B can also be calculated in the frequency domain using Theorem A.3. Transforming the system to row reduced form gives that

$$U(s) = \begin{bmatrix} -\frac{1}{J_1} & -\frac{1}{J_2} & 0 & s \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{1}{J_1} & -\frac{1}{J_2} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{U_0} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{U_1} s \tag{32}$$

and that

$$D(s) = \begin{bmatrix} 0 & 0 & \frac{1}{J_1} & -\frac{1}{J_2} \\ 0 & J_2 s & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \end{bmatrix} \tag{33}$$

with notation from section 2.2.

This gives that the row degrees of $D(s)$ are $r_1[D] = 0$, $r_2[D] = 1$, $r_3[D] = 0$, and $r_4[D] = 0$. Theorem A.3 thus gives that the transfer function is proper if and only if

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} B = 0. \tag{34}$$

What (34) says is that the last row of B must be zero, which is the same conclusion as was reached using the time domain method, Theorem A.1.

5 Discrete Time Linear Descriptor Systems

The discrete linear time invariant descriptor system is an equation on the form

$$Ex[k+1] + Fx[k] = Bw[k], \tag{35a}$$
$$y[k] = Cx[k] + e[k], \tag{35b}$$

where E, F, and C are constant matrices and $w[k]$ and $e[k]$ are white noise sequences, i.e., sequences of independent and identically distributed random variables. For this case it is possible to make the same transformation as for a continuous differential-algebraic equation if $\det[Ez + F]$ is not identically zero as a function of $z \in \mathbb{R}$ (Section 2) since the structure is similar. Similarly to the continuous time case, $x[k]$ will not be uniquely determined by $w(k)$ if $\det[Ez + F]$ is identically zero. A certain transformation

$$PEQx[k+1] + PFQx[k] = PBw[k] \tag{36}$$

with nonsingular matrices P and Q will thus give us the form

$$\begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} z_1[k+1] \\ z_2[k+1] \end{bmatrix} + \begin{bmatrix} -A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} z_1[k] \\ z_2[k] \end{bmatrix} = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} w[k]. \tag{37}$$

As in the continuous time case, we can write (37) in the form

$$z_1[k+1] = Az_1[k] + G_1 w[k] \tag{38a}$$
$$z_2[k] = \sum_{i=0}^{n-1} (-N)^i G_2 w[k+i]. \tag{38b}$$

The system (35) is thus well defined for all B-matrices, since no derivatives occur in this case. However, $z_2[k]$ will depend on future values of the noise. To avoid this, the noise sequence can be time shifted. If we let $\tilde{w}[k] = w[k+n-1]$ we can rewrite (38) according to

$$z_1[k+1] = Az_1[k] + G_1 \tilde{w}[k-n+1] \tag{39a}$$
$$z_2[k] = \sum_{i=-n+1}^{0} (-N)^i G_2 \tilde{w}[k+i] \tag{39b}$$

which can be transformed to a normal state-space description. This state-space description can then be used to implement a Kalman filter, which is discussed in Dai (1987). Other approaches to Kalman filtering of discrete-time linear descriptor systems are discussed in, Dai (1989a); Darouach *et al.* (1993); Deng and Liu (1999); Nikoukhah *et al.* (1998, 1999).

The sequences $w[k]$ and $\tilde{w}[k]$ will have the same statistical properties since they both are white noise sequences.

It can be also be noted that the same requirement that was put on B in the continuous time case may also be used in the discrete time case. This would then guarantee that the system would not depend on future noise values and the noise sequence would not have to be time shifted.

5.1 Frequency Domain

The ideas of time shifting the noise might become clearer if they are treated in the frequency domain. If we transform (35) to the frequency domain we get

$$X(z) = \underbrace{(Ez + F)^{-1}B}_{H(z)} W(z). \tag{40}$$

The only difference from a transfer function for a state-space system is that here $H(z)$ is noncausal in the general case. If we rewrite (40) as

$$X(z) = \underbrace{H(z)z^{-T}}_{\tilde{H}(z)} \underbrace{z^T W(z)}_{\tilde{W}(z)}, \tag{41}$$

then $\tilde{W}(z)$ will be a time shifted version of $W(z)$ and $\tilde{H}(z)$ will be a causal transfer function if T is large enough.

6 Conclusions

We have in this article proposed a framework for modeling and filtering of systems composed of linear differential-algebraic equations. The main reason for studying these systems is that they occur as the natural description delivered from object-oriented modeling software. At the core of this problem we find the question of how to incorporate stochastics into linear differential-algebraic equations. This has been solved in this paper in the case where white noise is used. The result was presented as two equivalent theorems, one in the time domain and one in the frequency domain. The resulting model fits into the optimal filtering framework and standard methods such as the Kalman filter applies. An example was also given, which showed that the conditions derived for how the noise can enter the system gives requirements which are physically motivated.

7 Acknowledgments

The authors would like to thank the Swedish Research Council and the Foundation for Strategic Research for supporting this work.

References

Åström, K. (1970). *Introduction to Stochastic Control Theory*. Mathematics in Science and Engineering. Academic Press, New York and London.

Brenan, K., Campbell, S., and Petzold, L. (1996). *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM's Classics in Applied Methematics. SIAM, New York.

Campbell, S. (1990). Descriptor systems in the 90's. In *proceedings of the 29th Conference on Decision and control*, pages 442–447, Honolulu, Hawaii, USA. IEEE.

Dai, L. (1987). State estimation schemes for singular systems. In *Preprints of the 10th IFAC World Congress, Munich, Germany*, volume 9, pages 211–215.

Dai, L. (1989a). Filtering and LQG problems for discrete-time stochastic singular systems. *IEEE Transactions on Automatic Control*, 34(10):1105–1108.

Dai, L. (1989b). *Singular Control Systems*. Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, New York.

Darouach, M., Zasadzinski, M., and Mehdi, D. (1993). State estimation of stochastic singular linear systems. *International Journal of Systems Sciences*, 24(2):345–354.

Deng, Z. and Liu, Y. (1999). Descriptor Kalman filtering. *International Journal of Systems Science*, 30(11):1205–1212.

Gerdin, M., Glad, T., and Ljung, L. (2003). Parameter estimation in linear differential-algebraic equations. In *proceedings of the 13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands. IFAC.

Glad, T. and Ljung, L. (2000). *Control Theory, Multivariable and Nonlinear Methods*. Taylor and Francis.

Gustafsson, F. (2000). *Adaptive Filtering and Change Detection*. John Wiley & Sons.

Kailath, T. (1980). *Linear systems*. Prentice Hall, Englewood Cliffs, N.J.

Ljung, L. and Glad, T. (2003). *Modellygge och simulering*. Studentlitteratur. To appear. In Swedish.

Mattsson, S., Elmqvist, H., and Otter, M. (1998). Physical system modeling with modelica. *Control Engineering Practice*, 6:501–510.

Nikoukhah, R., Campbell, S., 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, Florida, USA.

Nikoukhah, R., Campbell, S., and Delebecque, F. (1999). Kalman filtering for general discrete-time linear systems. *IEEE Transactions on Automatic Control*, 44(10):1829–1839.

Rugh, W. (1996). *Linear System Theory*. Information and system sciences series. Prentice Hall, Upper Saddle River, New Jersey, 2 edition.

Schein, O. and Denk, G. (1998). Numerical solution of stochastic differential-algebraic equations with applications to transient noise simulation of microelectronic circuits. *Journal of Computational and Applied Mathematics*, 100:77–92.

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

Tiller, M. (2001). *Introduction to Physical Modeling with Modelica*. Kluwer Academic Press, Boston.

Varga, A. (1992). Numerical algorithms and software tools for analysis and modelling of descriptor systems. In *proceedings of 2nd IFAC Workshop on System Structure and Control, Prague, Czechoslovakia*, pages 392–395.

Winkler, R. (2003). Stochastic differential algebraic equations of index 1 and applications in circuit simulation. *Journal of Computational and Applied Mathematics*, pages 477–505.

B

A Note on State Estimation as a Convex Optimization Problem

Authors: Thomas Schön, Fredrik Gustafsson, and Anders Hansson.

Edited version of paper originally published in *proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 61–64, Hong Kong, 2003.

A Note on State Estimation as a Convex Optimization Problem

Thomas Schön, Fredrik Gustafsson, and Anders Hansson.

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

Abstract

The Kalman filter computes the maximum *a posteriori* (MAP) estimate of the states for linear state-space models with Gaussian noise. We interpret the Kalman filter as the solution to a convex optimization problem, and show that we can generalize the MAP state estimator to any noise with log-concave density function and any combination of linear equality and convex inequality constraints on the states. We illustrate the principle on a hidden Markov model, where the state vector contains probabilities that are positive and sum to one.

Keywords: State estimation, Kalman filter, Convex optimization, Hidden Markov models, Linear regression.

1 Introduction

State estimation in stochastic linear models is an important problem in many model based approaches in signal processing and automatic control applications, where the Kalman filter is the standard method. However, if we have prior information of some kind it is often impossible to incorporate this in the Kalman filter framework. We will in this paper show how we can use prior information by considering the optimization problem that the Kalman filter solves. A similar treatment can be found in Robertson and Lee (2002), however they only consider quadratic problems, whereas we will consider a larger class of convex problems.

2 Convex Optimization

In this section we will give a very brief introduction to convex optimization (see also Vandenberghe and Boyd, 2001).

The main message in convex optimization is that one should *not* differ between linear and nonlinear optimization problems, but instead between convex and non-convex problems. This is due to the fact that the class of convex problems is much larger than that covered by linear problems, and we know that for a convex problem any local optimum is also a global optimum. Moreover, there exist efficient algorithms for solving convex optimization problems. A convex optimization problem is defined as

$$\begin{array}{lrcll} \min\limits_{x} & f_0(x) & & & \\ \text{s.t.} & f_i(x) & \leq & 0, & i=0,\ldots,m \\ & a_i^T x & = & b_i, & i=0,\ldots,n \end{array} \tag{1}$$

where the functions $f_0,\ldots,f_m$ are convex and the equality constraints are linear. We will in the following sections try to identify some estimation problems that can be cast as convex optimization problems.

3 Notation and Background

Maximum *a posteriori* (MAP) estimation (Jazwinski, 1970) is about finding an estimator of a stochastic variable z that maximizes the conditional density $p(z|y)$, given the observation y ($y \in \mathbb{R}^{n_y}$ and $z \in \mathbb{R}^{n_z}$). Thus, the MAP problem is

$$\max_{z} \quad \log(p(z|y)) \tag{2}$$

In the sequel, the measurement vectors y_i from time 0 to time t will be denoted $y_{0:t}$, and similarly $z_{0:t}$ denotes all unknowns including the initial values. The operator $z_i^{(j)}$ extracts the $j th$ element from the vector z_i.

The assumptions commonly used in the literature are that the elements in the z vectors are spatially and temporally independent ("white noise") and Gaussian distributed. We will insist on the independence assumption, but not on the assumption of Gaussian densities, giving us the following form of $\log(p(z))$ (suppressing the dependence on y)

$$\log(p(z_{0:t})) = \log(\prod_{i=0}^{t} p_{z_i}(z_i)) = \sum_{i=0}^{t} \log(p_{z_i}(z_i)). \tag{3}$$

Depending on the distribution, the objective function in (1) can be explicitely written as in Table 1 (see also Vandenberghe and Boyd, 2001).

4 Convex Optimization Estimation

In this section we will discuss the estimation problem in the presence of constraints. In Table 1 the objective functions are given for several log-concave[1] densities. Constraints arise in the derivation of some of these probability density functions (PDF),

[1] A function $f : \mathbb{R}^n \to \mathbb{R}$ is *log-concave* if $f(x) > 0$ for all x in the domain of f, and $\log(f)$ is a concave function (Vandenberghe and Boyd, 2001).

Table 1 *Objective functions in (1) for different normalized (zero mean and unit covariance) probability density functions.*

PDF	Objective function	Extra constraints
Gaussian	$\sum_{i=0}^{t} \|z_i\|^2$	
Exponential	$\sum_{i=0}^{t} \sum_{j=1}^{n_z} z_i^{(j)} - 1$	$z \geq 0$
Laplacian	$\sum_{i=0}^{t} \sum_{j=1}^{n_z} \|z_i^{(j)}\|$	
Uniform	constant	$-\sqrt{3} \leq z \leq \sqrt{3}$

but constraints also arise from prior information of some kind, e.g., a model assumption. This will be discussed in Section 6.

Assume we want to estimate $[x^T, z^T]^T$, where z has a certain known distribution, and that x and z are related through the constraints

$$A \begin{bmatrix} x \\ z \end{bmatrix} = b, \tag{4}$$

If we now want to use (2) we are faced with the problem of finding the joint distribution of x and z, which can be quite tedious.

Problem B.1 (Convex optimization estimation)
Assume that $p(z)$ is a known log-concave probability density function. The MAP-estimate for $[x^T, z^T]^T$, where x and z are related via (4) is given by

$$\begin{aligned} \max_{x,z} \quad & \log(p_z(z)) \\ \text{s.t.} \quad & A \begin{bmatrix} x \\ z \end{bmatrix} = b \end{aligned} \tag{5}$$

Remark: Any linear equalities and convex inequalities may be added to this formulation, and standard software applies.

This approach to estimation is presented in Vandenberghe and Boyd (2001). The standard estimation problem is to interpret x as the parameters conditioned on the measurements $x|y$, and then z is just a nuisance parameter. The standard approach, not often written explicitly, is to marginalize the nuisance parameters to get $p(x|y) = \int p(x|y,z)p(z|y)dz$ where the constraints are used explicitly. This works fine in a range of applications, and the solution most often has a quite simple form. In the general case, we can formulate the problem below.

5 Linear Regression Example

As an example of estimation, consider a linear regression problem in matrix form

$$Y = \Phi^T \theta + E. \tag{6}$$

Interpret $E \leftrightarrow z$ as a Gaussian nuisance parameter with variance σ^2, the regression parameter $\theta \leftrightarrow x$ as the parameter and $Y, \Phi \leftrightarrow y$ as the observations. The well-known result from marginalization is that

$$\theta \in \mathcal{N}((\Phi\Phi^T)^{-1}\Phi Y, \sigma^2(\Phi\Phi^T)^{-1}). \tag{7}$$

Alternatively, we can pose the problem as

$$\begin{aligned} \max_{x,\, z} \quad & \log(p_E(E)) \\ \text{s.t.} \quad & \begin{bmatrix} \Phi^T & \mathbf{1} \end{bmatrix} \begin{bmatrix} \theta \\ E \end{bmatrix} = Y \end{aligned} \tag{8}$$

If this regression model happens to be an ARX model of a transfer function

$$G(e^{i\omega}) = \frac{\sum_l b^{(l)} e^{-i\omega l}}{1 + \sum_l a^{(l)} e^{-i\omega l}}, \tag{9}$$

in system identification, we use $\theta = [a^T,\, b^T]^T$. Now, we can simply add constraints such as bounded DC gain $L \leq G(0) \leq U$, or more generally, any lower and upper bound on the transfer function

$$L(\omega) \leq \frac{\sum_l b^{(l)} e^{-i\omega l}}{1 + \sum_l a^{(l)} e^{-i\omega l}} \leq U(\omega), \tag{10}$$

which is easily rewritten in the standard form. Similarly, any other interval for any other frequency of the transfer function can be bounded.

6 Convex Optimization Filtering

In Section 4 we talked about constraints in general. We will in this section discuss a special type of constraints, namely the ones that appear in describing the dynamic behavior of a model. In order to obtain convex problems we will use linear models of the dynamics. The following model

$$Ex_{t+1} = Ax_t + Bw_t + Ke_t, \tag{11a}$$

$$y_t = Cx_t + De_t, \tag{11b}$$

together with a density for the initial state, p_{x_0}, and $p_{e_t}(e_t)$, $p_{w_t}(w_t)$ will constitute our model. With $E = I$, $K = 0$ we have the standard state-space model, and with $E = I$, $B = 0$, $D = I$ we have the so called *innovation form*. If the E-matrix in (11a) is invertible we can rewrite the equation in a state-space model. Otherwise we have what is commonly referred to as a *descriptor model* (Luenberger, 1977).

To put state filtering in the general estimation form as in Problem B.1, let

$$z = \begin{bmatrix} x_0^T & w_{0:t-1}^T & e_{0:t}^T \end{bmatrix}^T, \tag{12}$$

and interpret x as $x_{1:t}|y_{1:t}$. To rewrite the conditional density more explicitly, use the independence assumption and (3), which gives

$$\log(p(x_0, w_{0:t-1}, e_{0:t})) = \log(p_{x_0}(x_0)) + \sum_{i=0}^{t-1} \log(p_{w_i}(w_i)) + \sum_{i=0}^{t} \log(p_{e_i}(e_i)). \quad (13)$$

Using Bayes' theorem , $p(z|y) = p(y|z)p(z)/p(y)$ and the fact that

$$p(x_t) = p_{x_0}(x_0) \prod_{i=0}^{t-1} p_{w_i}(w_i), \quad (14a)$$

$$p(y_t|x_t) = \prod_{i=0}^{t} p_{e_i}(e_i), \quad (14b)$$

we obtain the following objective function

$$p(x_0, w_{0:t-1}, e_{0:t}) = \prod_{i=0}^{t} p_{e_i}(e_i) p_{x_0}(x_0) \prod_{i=0}^{t-1} p_{w_i}(w_i). \quad (15)$$

Conditioned on z in (12), the states in (11) are uniquely defined by a deterministic mapping $x = f(z)$, which implies that $p(x|z) = f(z)$ contains nothing stochastic. That is, the MAP-estimate of x and z are simply related by $\hat{x}^{MAP} = f(\hat{z}^{MAP})$. Similarly, the joint MAP-estimate x, z in the convex optimization formulation is given by maximizing $p(z)$, since $p(z, x) = p(x|z)p(z) = f(z)p(z)$ by Bayes' theorem. Hence we have now justified the following general convex filtering problem.

Problem B.2 (Convex optimization filtering)
Assume that the probability density functions $p_{x_0}(x_0)$, $p_{w_i}(w_i)$, and $p_{e_i}(e_i)$ are log-concave. In the presence of constraints in terms of a dynamic model (11) the MAP-estimate is the solution $\hat{x}_t = x_t$ to the following problem

$$\begin{aligned} \max_{x_{0:t}, z} \quad & \log(p_{x_0}(x_0)) + \sum_{i=0}^{t-1} \log(p_{w_i}(w_i)) + \sum_{i=0}^{t} \log(p_{e_i}(e_i)) \\ \text{s.t.} \quad & E_i x_{i+1} = A_i x_i + B_i w_i + K_i e_i, \qquad i = 0, \ldots, t-1 \\ & y_i = C_i x_i + D_i e_i, \qquad i = 0, \ldots, t \end{aligned}$$

Remark: Any linear equalities and convex inequalities may be added to this formulation, and standard software applies.

As is evident from Problem B.2 we see that we are free to use different densities for the different disturbances $p_{x_0}(x_0), p_{w_i}(w_i)$, and $p_{e_i}(e_i)$. It is here also worth noting that the recursive solution to Problem B.2 under the assumptions of Gaussian densities and a nonsingular E-matrix is the celebrated Kalman filter. This has been known for a long time (see e.g., Kailath, 1974; Sorenson, 1970) for nice historical accounts of this fact, and for a proof see Rao (2000). It is also worthwhile noting that Problem B.2 under the assumption of Gaussian disturbances is

a weighted least squares problem. To see this combine B.2 and the Gaussian case in Table 1, where the weights are the inverse of the covariance matrices. This provides a deterministic interpretation of the problem that the Kalman filter solves. For more on the similarities and differences between deterministic and stochastic filtering see e.g., Kailath *et al.* (2000). We also see that if we solve Problem B.2 we will not only obtain the filtered estimate $\hat{x}_{t|t}$, but also all the smoothed estimates, $\hat{x}_{i|t}, i = 0, \ldots, t-1$.

So why should we solve the estimation problem via B.2, which demands more computations, instead of via the Kalman filter? There are two reasons. The first reason is that we can handle all log-concave density functions, not just the Gaussian. The second reason is that we can add any prior information, in convex form, to problem B.2. That is we can add linear equality constraints and convex inequality constraints, and still find the optimal estimate. We will see an illustration of this in the example in the subsequent section.

7 HMM Example

There are mainly two filtering problems, where there exist finite-dimensional recursive optimal filters, and in particular a finite-dimensional MAP-estimator. One is, as already mentioned, linear state-space models with Gaussian noise. Here the Kalman filter is optimal in ML, MAP and minimum variance senses. For non-Gaussian noises, the Kalman filter computes the linear state estimate with minimum variance, but it is no longer the MAP or ML estimator.

The other case is hidden Markov models (HMM). Interestingly, it has been pointed out Andersson (2002) that the HMM can be written as a state-space model. That is, the Kalman filter computes the best possible linear estimate of the Markov state. This fact makes it possible to compare conceptually different approaches on the same example!

A hidden Markov model is defined by a discrete variable $\xi \in (1, 2, \ldots, n)$ with a known transition probability matrix A, where $A^{(i,j)} = \Pr(\xi_t = i | \xi_{t-1} = j)$, that is, given that $\xi = j$ at time $t-1$, the probability that $\xi = i$ at time t is $A^{(i,j)}$. We will assume an observation process $\nu \in (1, 2, \ldots, m)$, where $\Pr(\nu = i | \xi = j) = C^{(i,j)}$. The filter for computing the *a posteriori* probabilities can be expressed as the recursion

$$\pi_t^{(i)} = p(\xi_t = i) = \frac{\sum_{j=1}^{n} \sum \pi_{t-1}^{(j)} A^{(i,j)} C^{(\nu_t, j)}}{\sum_{j=1}^{n} \sum \pi_{t-1}^{(j)} C^{(\nu_t, j)}}. \tag{16a}$$

The MAP-estimate is $\hat{\xi}_t = \arg\max_i \pi_t^{(i)}$. Now, the HMM can be written as the state-space model

$$x_{t+1} = Ax_t + w_t, \tag{17a}$$

$$y_t = Cx_t + e_t, \tag{17b}$$

where $x_t^{(i)} = \Pr(\xi_t = i)$ and $y_t^{(i)} = \Pr(\nu_t = i)$. This is the state-space form (11) with $B = D = E = I, K = 0$) where the disturbances are zero-mean white noises, and the stationary covariance matrices can be shown to be

$$Q = \mathrm{Cov}[w_t] = \mathrm{diag}(\pi) - A\mathrm{diag}(\pi)A^T, \tag{18a}$$

$$R = \mathrm{Cov}[e_t] = \mathrm{diag}(C\pi) - C\mathrm{diag}(\pi)C^T, \tag{18b}$$

where π is the stationary solution to (in vector form)

$$\pi = \lim_{t\to\infty} A^t \pi_0, \qquad \text{where} \qquad \pi_0 > 0. \tag{19}$$

Since the states x we are estimating in a HMM are probabilities we have the following prior information on the states

$$\sum_{i=1}^{2} x^{(i)} = 1, \quad \text{and} \quad x^{(i)} \geq 0,\ i = 1, 2. \tag{20}$$

In the standard Kalman filter it is impossible to incorporate this prior information about the states, however in Problem B.2 it is straightforward. We will now examine four different filters using an increasing amount of prior information (in 1-3 we have approximated w_t and e_t in (17) as Gaussian with zero-mean and covariances (18)):

1. The Kalman filter.
2. The convex optimization filter with constraint $\sum_i x_t^{(i)} = 1$. This case can alternatively be computed by the Kalman filter using $P_0 = p_0 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ and any $\sum_i x_0^{(i)} = 1$, or by using the fictitious measurement $y_0 = [1, 1, \ldots, 1]x_0 = 1$ with zero measurement noise. Note, however, that the Ricatti equation will be singular here, which may imply certain numerical difficulties. A more theoretically sound alternative is given in Andersson (2002).
3. The convex optimization filter with constraint (20).
4. The optimal filter (16).

Table 2 *RMSE values for the different filters.*

1. Kalman filter	0.585
2. B.2 with $x_1 + x_2 = 1$	0.573
3. B.2 with $x_1 + x_2 = 1$ and $x \geq 0$	0.566
4. Optimal filter	0.403

The numerical example is taken from Andersson (2002), where

$$A = C = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix} \tag{21}$$

In Table 2, the root mean square error (RMSE) is given for these four cases and in Figure 1 the states are shown. From this table it is obvious that we can obtain better estimates by using more information in this case. Of course, the convex optimization filters cannot compare to the performance of the optimal filter. However, the point is to show the flexibility of the approach, and the problem of consideration can be generalized with more constraints or a more complicated measurement relation, such that the optimal filter does no longer exist.

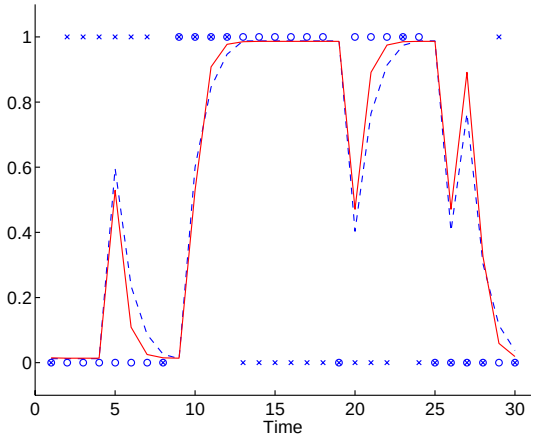


Figure 1 *The true state is marked by o, and the measured states by x. The dashed/solid line is the estimate from filter 3, respective 4.*

8 Conclusions

We have formulated the state estimation problem in a convex optimization framework. In this way, well-known numerical efficient algorithms can be used to compute the MAP-estimate of the state vector, without any problems with local minima. Compared to the Kalman filter, the advantage is that any log-concave noise densities can be used and any linear equality or convex inequality constraints may be included, while the main drawback is that no recursive convex optimization algorithm is yet available, which makes the approach computer intensive.

References

Andersson, S. (2002). *Hidden Markov Models - Traffic Modeling and Subspace Methods*. PhD thesis, Centrum for Mathematical Sciences, Mathematical Statistics, Lund University, Lund, Sweden.

Jazwinski, A. (1970). *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York.

Kailath, T. (1974). A view of three decades of linear filtering theory. *IEEE Transactions on Information Theory*, IT-20(2):146–181.

Kailath, T., Sayed, A., and Hassibi, B. (2000). *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, New Jersey.

Luenberger, D. (1977). Dynamic equations in descriptor form. *IEEE Transactions on Automatic Control*, AC-22(3).

Rao, C. (2000). *Moving Horizon Strategies for the Constrained Monitoring and Control of Nonlinear Discrete-Time Systems*. PhD thesis, University of Wisconsin Madison.

Robertson, D. and Lee, J. (2002). On the use of constraints in least squares estimation and control. *Automatica*, 38:1113–1123.

Sorenson, H. (1970). Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum*, 7:63–68.

Vandenberghe, L. and Boyd, S. (2001). Convex optimization. To be published.

C

Marginalized Particle Filters for Nonlinear State-space Models

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

Submitted to *IEEE Transactions on Signal Processing*,

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

Marginalized Particle Filters for Nonlinear State-space Models

Thomas Schön and Fredrik Gustafsson

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

Per-Johan Nordlund

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

Abstract

The recently developed particle filter offers a general numerical tool to approximate the state *a posteriori* density in nonlinear and non-Gaussian filtering problems with arbitrary accuracy. Because the particle filter is fairly easy to implement and tune, it has quickly become a popular tool in signal processing applications. Its main drawback is that it is quite computer intensive. For a given filtering accuracy, the computational complexity increases quickly with the state dimension. One remedy to this problem is what in statistics is called Rao-Blackwellization, where states appearing linearly in the dynamics are marginalized out. This leads to that a Kalman filter is attached to each particle. Our main contribution here is to sort out when marginalization is possible for state space models, and to point out the implications in some typical signal processing applications. The methodology and impact in practice is illustrated on terrain navigation for aircrafts. The marginalized particle filter for a state-space model with nine states is evaluated on real aircraft data, and the result is that very good accuracy is achieved with quite reasonable complexity.

Keywords: State estimation, Particle filter, Kalman filter, Navigation systems, Nonlinear systems.

1 Introduction

The nonlinear non-Gaussian filtering problem we consider consists of computing the *a posteriori* density of the state vector, given the observed measurements, in

a general discrete-time state-space model, where a general formulation appears in Model 1 below.

Model 1

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

where y_t is the measurement at time t, x_t is the state, w_t is the state noise and e_t is the measurement noise. The two noise densities have to be known and independent.

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

$$p(x_t|Y_t) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})}, \tag{2a}$$
$$p(y_t|Y_{t-1}) = \int_{\mathbb{R}^{n_x}} p(y_t|x_t)p(x_t|Y_{t-1})dx_t, \tag{2b}$$

and the following time recursion

$$p(x_{t+1}|Y_t) = \int_{\mathbb{R}^{n_x}} p(x_{t+1}|x_t)p(x_t|Y_t)dx_t, \tag{2c}$$

and the recursion is initiated by $p(x_0|Y_{-1}) = p(x_0)$. For linear Gaussian models, the integral can be solved analytically with a finite dimensional representation leading to the Kalman filter recursion, where the mean and covariance matrix of the state are propagated (Anderson and Moore, 1979). Generally, no finite dimensional representation of the *a posteriori* density exists. Therefore, numerical approximations of the integral have been proposed. A recent important contribution is to apply simulation based methods from mathematical statistics, the sequential Monte Carlo methods, commonly referred to as particle filters (see Doucet, 1998; Doucet *et al.*, 2001a; Gordon *et al.*, 1993).

A problem inherent in the particle filter is that in general it requires a lot of computational power. If there is a linear sub-structure in the state-space model (1) this can be utilized in order to obtain better estimates and possibly reduce the computational demands. The idea is to partition the state vector according to

$$x_t = \begin{bmatrix} x_t^l \\ x_t^n \end{bmatrix}, \tag{3}$$

where x_t^l denotes the state variable with conditional linear dynamics and x_t^n denotes the nonlinear state variable. Using Bayes' theorem we can then marginalize out the linear state variable from (1) and estimate them using a finite-dimensional optimal filter. The remaining nonlinear state variables are then estimated using the particle filter. This is sometimes referred to as Rao-Blackwellization (Doucet *et al.*,

2001b). This idea is certainly not a new one, it has been around for quite some time (see Andrieu and Doucet, 2002; Chen and Liu, 2000; Doucet *et al.*, 2001b). The contribution in this article is that we sort out the details in the case where the model class is a general nonlinear state-space model with a linear sub-structure. This model class is important in engineering applications, e.g., positioning, target tracking and collision avoidance (Gustafsson *et al.*, 2002). We provide an application example where the marginalized particle filter discussed in this article is used in an integrated aircraft navigation system. The algorithm has been tested and evaluated on *authentic flight data* from the Swedish fighter aircraft JAS 39 Gripen.

Section 2 is devoted to briefly explaining the standard particle filter according to Gordon *et al.* (1993). We will then explain the idea of using marginalization in conjunction with state-space models in three steps, in order to make the presentation easy to follow. This is done in Sections 3, and 4, and in Section 5 the most general state-space model is stated. We will then comment upon some important special cases and discuss some modeling issues in Section 6. Finally, the application example is given in Section 7 and the conclusions are stated in Section 8.

2 Introducing the Particle Filter

Before introducing the idea of the marginalized particle filter we briefly explain the standard particle filter. The particle filter provides an approximative solution for the problem of recursively estimating the *a posteriori* density function, $p(X_t|Y_t)$, for a nonlinear discrete-time model on the form (1). We will use X_t to denote the set of states up to time t, i.e., $\{X_i\}_{i=0}^{t}$ and Y_t is defined analogously. We are mostly interested in one of the marginals to the *a posteriori* density, the filtering density, $p(x_t|Y_t)$. This density is approximated using a large set of samples (also called particles, hence the name particle filter), $\{x_t^{(i)}\}_{i=1}^{N}$, according to

$$\hat{p}_N(x_t|Y_t) = \sum_{i=1}^{N} \tilde{q}_t^{(i)} \delta(x_t - x_t^{(i)}), \tag{4}$$

where $\delta(\cdot)$ is Dirac's delta function and $\tilde{q}_t^{(i)}$ are the normalized importance weights, which typically are updated according to

$$q_t^{(i)} = p(y_t|x_t^{(i)}) q_{t-1}^{(i)}. \tag{5}$$

This means that the most likely samples, i.e., the samples that correspond to a large likelihood will be assigned a large weight. The key-step, which made the particle filter work in practice was the resampling step, introduced by Gordon *et al.* (1993), based on the weighted bootstrap in Smith and Gelfand (1992). The entire particle filtering algorithm is given below. See e.g., (Doucet, 1998; Doucet *et al.*, 2001a) for a thorough introduction to the particle filter.

Algorithm C.1 (The particle filter)

1. *Initialization: For* $i = 1, \ldots, N$, *initialize the particles,* $x^{(i)}_{0|-1} \sim p_{x_0}(x_0)$.
2. *For* $i = 1, \ldots, N$, *evaluate the importance weights* $q^{(i)}_t = p(y_t|x^{(i)}_{t|t-1})$ *and normalize* $\tilde{q}^{(i)}_t = \frac{q^{(i)}_t}{\sum_{j=1}^N q^{(j)}_t}$.
3. *Measurement update: Resample with replacement* N *particles according to*

$$\Pr(x^{(i)}_{t|t} = x^{(j)}_{t|t-1}) = \tilde{q}^{(j)}_t \tag{6}$$

4. *Time update: For* $i = 1, \ldots, N$, *predict new particles according to*

$$x^{(i)}_{t+1|t} \sim p(x_{t+1|t}|x^{(i)}_{t|t}) \tag{7}$$

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

Note that we have used a double index in the algorithm above, $x_{t+1|t}$ means the prediction of x at time $t+1$ given the information available at time t. This is to make the comparison with the Kalman filter easier. There exist many alternative particle filter algorithms. We have given the simplest algorithm here. However, most of what is said in this article applies analogously to other particle filter algorithm as well.

In the subsequent section we will explain how to use marginalization in nonlinear state-space models. We do this in three steps in order to communicate the idea as clearly as possible. In each of these steps we use more and more advanced model classes.

3 Variance Reduction by Marginalization

We will now start explaining the ideas of the marginalized particle filter using the following model class (the gaps in this model are placed there intentionally, in order to facilitate an easier comparison to the general model (23)),

Model 2

$$x^n_{t+1} = f^n_t(x^n_t) + w^n_t, \tag{8a}$$
$$x^l_{t+1} = A^l_t(x^n_t)x^l_t + w^l_t, \tag{8b}$$
$$y_t = h_t(x^n_t) + C_t(x^n_t)x^l_t + e_t, \tag{8c}$$

where we assume that the state noise is white and Gaussian distributed according to

$$w_t = \begin{bmatrix} w_t^l \\ w_t^n \end{bmatrix} \sim \mathcal{N}(0, Q_t), \quad Q_t = \begin{bmatrix} Q_t^l & 0 \\ 0 & Q_t^n \end{bmatrix}, \tag{9a}$$

and that the measurement noise is white and Gaussian distributed according to

$$e_t \sim \mathcal{N}(0, R_t). \tag{9b}$$

Furthermore we assume that x_0^l is white and Gaussian,

$$x_0^l \sim \mathcal{N}(\bar{x}_0, \bar{P}_0). \tag{9c}$$

The density of x_0^n can be arbitrary, but it has to be known.

As mentioned in the previous section the aim is to estimate $p(x_t|Y_t)$ recursively. This can of course be accomplished using the particle filter for the entire state vector, x_t. However, we can exploit the linear structure inherent in (8), to obtain better estimates, by using the optimal filter for the linear part. This can be done by analytically marginalizing out the linear variables from $p(x_t|Y_t)$. Using Bayes' theorem we obtain

$$p(x_t^l, X_t^n|Y_t) = \underbrace{p(x_t^l|X_t^n, Y_t)}_{\text{Optimal KF}} \underbrace{p(X_t^n|Y_t)}_{\text{Approximate PF}}, \tag{10}$$

where $p(x_t^l|X_t^n, Y_t)$ is analytically tractable, it is given by the Kalman filter (KF), see Lemma C.1 below. Furthermore $p(X_t^n|Y_t)$ can be estimated using the particle filter (PF). This will intuitively provide better estimates for a given number of particles as compared to the standard particle filter. The reason is that the dimension of $p(x_t^n|Y_t)$ is smaller than the dimension of $p(x_t^l, x_t^n|Y_t)$, implying that the particles live in a smaller space. Theoretical justification of this intuition is provided in e.g., Doucet *et al.* (1999). Before we state Lemma C.1 we will clarify a notational matter. When we write

$$p(x) = \mathcal{N}(m, P), \tag{11}$$

we mean

$$p(x) = \frac{1}{(2\pi)^{n_x/2}(\det[P])^{1/2}} e^{-\frac{1}{2}(x-m)^T P^{-1}(x-m)}, \tag{12}$$

i.e., that the underlying stochastic variable, x, is distributed according to the normal distribution, with expectation m and covariance P. We use (11) instead of (12) to obtain a clearer and more compact notation. For the sake of brevity we suppress the dependence of x_t^n in A_t, C_t, and h_t below.

Lemma C.1
Given Model 2, the conditional probability density functions for $x^l_{t|t}$ and $x^l_{t+1|t}$ are given by

$$p(x^l_t|X^n_t, Y_t) = \mathcal{N}(\hat{x}^l_{t|t}, P_{t|t}), \tag{13a}$$
$$p(x^l_{t+1}|X^n_{t+1}, Y_t) = \mathcal{N}(\hat{x}^l_{t+1|t}, P_{t+1|t}), \tag{13b}$$

where

$$\hat{x}^l_{t|t} = \hat{x}^l_{t|t-1} + K_t(y_t - h_t - C_t\hat{x}^l_{t|t-1}), \tag{14a}$$
$$P_{t|t} = P_{t|t-1} - K_tC_tP_{t|t-1}, \tag{14b}$$
$$S_t = C_tP_{t|t-1}C^T_t + R_t, \tag{14c}$$
$$K_t = P_{t|t-1}C^T_tS^{-1}_t, \tag{14d}$$

and

$$\hat{x}^l_{t+1|t} = A^l_t\hat{x}^l_{t|t}, \tag{15a}$$
$$P_{t+1|t} = A^l_tP_{t|t}(A^l_t)^T + Q^l_t. \tag{15b}$$

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

Proof See Appendix A for the proof. □

The second density, $p(X^n_t|Y_t)$, in (10) will be approximated using the particle filter as mentioned above. In order to see how this is done we can write $p(X^n_t|Y_t)$ as

$$p(X^n_t|Y_t) = \frac{p(y_t|X^n_t, Y_{t-1})p(x^n_t|X^n_{t-1}, Y_{t-1})}{p(y_t|Y_{t-1})}p(X^n_{t-1}|Y_{t-1}), \tag{16}$$

where an approximation of $p(X^n_{t-1}|Y_{t-1})$ is provided by the previous iteration of the particle filter. In order for the particle filter to perform the update (16) we need analytical expressions for $p(y_t|X^n_t, Y_{t-1})$ and $p(x^n_t|X^n_{t-1}, Y_{t-1})$. They are provided by the following lemma.

Lemma C.2
For Model 2 we have that

$$p(y_t|X^n_t, Y_{t-1}) = \mathcal{N}(h_t + C_t\hat{x}^l_{t|t-1},\, C_tP_{t|t-1}C^T_t + R_t), \tag{17a}$$
$$p(x^n_{t+1}|X^n_t, Y_t) = \mathcal{N}(f^n_t,\, Q^n_t). \tag{17b}$$

Proof See Appendix B. □

Hence, for each particle, $\{x^{n,(i)}_t\}^N_{i=1}$, we form the corresponding linear system (8b) – (8c) and estimate the linear states using the Kalman filter. Hence, there is one Kalman filter associated with each particle. Finally, the overall algorithm for estimating the states in the model class (8) is given below.

Algorithm C.2 (The marginalized particle filter for Model 2)

1. *Initialization: For* $i = 1, \ldots, N$, *initialize the particles,* $x_{0|-1}^{n,(i)} \sim p_{x_0^n}(x_0^n)$ *and set* $\{x_{0|-1}^{l,(i)}, P_{0|-1}^{(i)}\} = \{\bar{x}_0^l, \bar{P}_0\}$.
2. *For* $i = 1, \ldots, N$, *evaluate the importance weights* $q_t^{(i)} = p(y_t|X_t^{n,(i)}, Y_{t-1})$ *according to (17a) and normalize* $\tilde{q}_t^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^N q_t^{(j)}}$.
3. *Particle filter measurement update: Resample with replacement* N *particles according to,*
$$Pr(x_{t|t}^{n,(i)} = x_{t|t-1}^{n,(j)}) = \tilde{q}_t^{(j)}.$$
4. *Particle filter time update and Kalman filter*
   (a) *Kalman filter measurement update, using (14).*
   (b) *Particle filter time update: For* $i = 1, \ldots, N$, *predict new particles using (17b) according to*
   $$x_{t+1|t}^{n,(i)} \sim p(x_{t+1|t}^n|X_t^{n,(i)}, Y_t).$$
   (c) *Kalman filter time update, using (15).*
5. *Set* $t := t + 1$ *and iterate from step 2.*

Now, the only difference from the standard particle filter is that the prediction stage has been changed. In the standard particle filter the prediction stage is given solely by step 4*b* in the algorithm given above. Hence, steps 4*a* and 4*c* do not exist in the standard particle filter. Here these steps take care of the estimation of the linear state variables. Step 4*a* is normally referred to as the *measurement update* in the Kalman filter. In step 4*b* we obtain a prediction of the nonlinear state, $x_{t+1|t}^n$, which, according to (8a) does not contain any information about the linear state variables. This means that we cannot use $x_{t+1|t}^n$ to obtain better estimates of the linear state variables in this case. In the model class discussed in the next section we will see that $x_{t+1|t}^n$ does indeed contain information about the linear state variables. The difference will occur in the time update equation in the Kalman filter, i.e., in (15). Finally, the estimates as expected means of the linear state variables and its covariances are given by (Nordlund, 2002)

$$\hat{x}_{t|t}^l = \sum_{i=1}^N \tilde{q}_t^{(i)} \hat{x}_{t|t}^{l,(i)} \approx E_{p(x_t^l|Y_t)}\left[x_t^l\right], \tag{18a}$$

$$\hat{P}_{t|t} = \sum_{i=1}^N \tilde{q}_t^{(i)} \left(P_{t|t}^{(i)} + (\hat{x}_{t|t}^{l,(i)} - \hat{x}_{t|t}^l)(\hat{x}_{t|t}^{l,(i)} - \hat{x}_{t|t}^l)^T\right) \tag{18b}$$

$$\approx E_{p(x_t^l|Y_t)}\left[\left((x_t^l)^2 - E_{p(x_t^l|Y_t)}\left[(x_t^l)^2\right]\right)^2\right]. \tag{18c}$$

where $\tilde{q}_t^{(i)}$ are the normalized importance weights, provided in step 2 in the algorithm above.

4 Extending the Model Class

We will in this section make the model class (8) a bit more general by adding the term $A_t^n(x_t^n)x_t^l$ to (8a) and see how that affects the estimation problem. Hence, we now have the following model class,

Model 3

$$x_{t+1}^n = f_t^n(x_t^n) + A_t^n(x_t^n)x_t^l + w_t^n, \tag{19a}$$
$$x_{t+1}^l = A_t^l(x_t^n)x_t^l + w_t^l, \tag{19b}$$
$$y_t = h_t(x_t^n) + C_t(x_t^n)x_t^l + e_t, \tag{19c}$$

with the same assumptions as in Model 2.

The difference with this in comparison to Model 2 is that it is no longer true that the nonlinear state at the next time instant, x_{t+1}^n, is independent of the linear state at the current time instant, x_t^l. This implies that there will be information about the linear state, x_t^l, in the prediction of the nonlinear state, $x_{t+1|t}^n$, given by the particle filter. This will lead to that the algorithm given in the previous section has to be changed. To understand the change let us now assume that step 4*b* has just been completed in Algorithm C.2. That means that the predictions, $x_{t+1|t}^n$, are available and the model can now be written (the information in the measurement, y_t, has already been used in step 4*a*)

$$x_{t+1}^l = A_t^l x_t^l + w_t^l, \tag{20a}$$
$$z_t = A_t^n x_t^l + w_t^n, \tag{20b}$$

where

$$z_t = x_{t+1}^n - f_t^n. \tag{20c}$$

Looking at these equations we see that we can interpret z_t as a measurement and w_t^n as the corresponding measurement noise. Since (20) is a linear model, with Gaussian noise the optimal state estimate is given by the Kalman filter, according to

$$\hat{x}_{t|t}^{l*} = \hat{x}_{t|t}^l + L_t(z_t - A_t^n \hat{x}_{t|t}^l), \tag{21a}$$
$$P_{t|t}^* = P_{t|t} - L_t N_t L_t^T, \tag{21b}$$
$$L_t = P_{t|t}(A_t^n)^T N_t^{-1}, \tag{21c}$$
$$N_t = A_t^n P_{t|t}(A_t^n)^T + Q_t^n, \tag{21d}$$

where we have used a star, $*$, to distinguish this second measurement update from the first one. Furthermore, $\hat{x}^l_{t|t}$, and $P_{t|t}$ are given by (14a) and (14b) respectively. Now, we have to merge this second measurement update with the time update in order to obtain the predicted states. This gives us

$$\hat{x}^l_{t+1|t} = A^l_t \hat{x}^l_{t|t} + L_t(z_t - A^n_t \hat{x}^l_{t|t}), \tag{22a}$$
$$P_{t+1|t} = A^l_t P_{t|t}(A^l_t)^T + Q^l_t - L_t N_t L^T_t, \tag{22b}$$
$$L_t = A^l_t P_{t|t}(A^n_t)^T N^{-1}_t, \tag{22c}$$
$$N_t = A^n_t P_{t|t}(A^n_t)^T + Q^n_t. \tag{22d}$$

For a formal proof of this fact the reader is referred to Appendix C.

Hence, the only thing that has to be changed in Algorithm C.2 for it to be valid for the more general Model 3 discussed in this section is that (15) is replaced by (22).

The second measurement update is labeled measurement update due to the fact that the mathematical structure of the equations are the same as a measurement update in the Kalman filter, but it is not a real measurement update, since there does not exist any new information. However, there is more information available in the prediction of the nonlinear state variables, $x^n_{t+t|t}$. The second measurement update can thus be thought of as a correction to the real measurement update, using the information provided by the prediction of the nonlinear state variables.

5 The General Case

A very general state-space model, where marginalization can be applied, is given by

Model 4

$$x^n_{t+1} = f^n_t(x^n_t) + A^n_t(x^n_t)x^l_t + G^n_t(x^n_t)w^n_t, \tag{23a}$$
$$x^l_{t+1} = f^l_t(x^n_t) + A^l_t(x^n_t)x^l_t + G^l_t(x^n_t)w^l_t, \tag{23b}$$
$$y_t = h_t(x^n_t) + C_t(x^n_t)x^l_t + e_t, \tag{23c}$$

where we assume that the state noise is white and Gaussian distributed with

$$w_t = \begin{bmatrix} w^l_t \\ w^n_t \end{bmatrix} \sim \mathcal{N}(0, Q_t), \quad Q_t = \begin{bmatrix} Q^l_t & Q^{ln}_t \\ (Q^{ln}_t)^T & Q^n_t \end{bmatrix}, \tag{24a}$$

and that the measurement noise is white and Gaussian distributed according to

$$e_t \sim \mathcal{N}(0, R_t). \tag{24b}$$

Furthermore, we assume that x^l_0 is white and Gaussian,

$$x^l_0 \sim \mathcal{N}(x_0, P_0). \tag{24c}$$

The density of x^n_0 can be arbitrary, but it has to be known.

In certain special cases some of these assumptions on the noises can be relaxed, we will discuss this issue more in the subsequent section.

Analogously to the previous sections, the filtering distribution, $p(x_t|Y_t)$ is split using Bayes' theorem according to

$$p(x_t^l, X_t^n|Y_t) = p(x_t^l|X_t^n, Y_t)p(X_t^n|Y_t). \tag{25}$$

The linear states are estimated using the same strategy as was discussed in the previous section. The three steps that have to be done are two measurement updates (one using the information available in y_t and one using the information, if any, available in $x_{t+t|t}^n$) and one time update. The following theorem explains how the linear states are estimated.

Theorem C.1
Using Model 4 the conditional probability density functions for $x_{t|t}^l$ and $x_{t+1|t}^l$ are given by

$$p(x_t^l|X_t^n, Y_t) = \mathcal{N}(\hat{x}_{t|t}^l, P_{t|t}), \tag{26a}$$
$$p(x_{t+1}^l|X_{t+1}^n, Y_t) = \mathcal{N}(\hat{x}_{t+1|t}^l, P_{t+1|t}), \tag{26b}$$

where

$$\hat{x}_{t|t}^l = \hat{x}_{t|t-1}^l + K_t(y_t - h_t - C_t\hat{x}_{t|t-1}^l), \tag{27a}$$
$$P_{t|t} = P_{t|t-1} - K_t M_t K_t^T, \tag{27b}$$
$$M_t = C_t P_{t|t-1} C_t^T + R_t, \tag{27c}$$
$$K_t = P_{t|t-1} C_t^T M_t^{-1}, \tag{27d}$$

and

$$\begin{aligned}\hat{x}_{t+1|t}^l = \bar{A}_t^l \hat{x}_{t|t}^l + G_t^l (Q_t^{ln})^T (G_t^n Q_t^n)^{-1} z_t \\ + f_t^l + L_t(z_t - A_t^n \hat{x}_{t|t}^l),\end{aligned} \tag{28a}$$
$$P_{t+1|t} = \bar{A}_t^l P_{t|t} (\bar{A}_t^l)^T + G_t^l \bar{Q}_t^l (G_t^l)^T - L_t N_t L_t^T, \tag{28b}$$
$$N_t = A_t^n P_{t|t} (A_t^n)^T + G_t^n Q_t^n (G_t^n)^T, \tag{28c}$$
$$L_t = \bar{A}_t^l P_{t|t} (A_t^n)^T N_t^{-1}, \tag{28d}$$

where we have defined

$$z_t = x_{t+1}^n - f_t^n, \tag{29a}$$
$$\bar{A}_t^l = A_t^l - G_t^l (Q_t^{ln})^T (G_t^n Q_t^n)^{-1} A_t^n, \tag{29b}$$
$$\bar{Q}_t^l = Q_t^l - (Q_t^{ln})^T (Q_t^n)^{-1} Q_t^{ln}. \tag{29c}$$

Proof See Appendix C. □

Note that (24a) differers from (9a). If $Q_t^{ln} = 0$ then $\bar{A}_t^l = A_t^l$ and $\bar{Q}_t^l = Q_t^l$. We have now taken care of the first density, $p(x_t^l|X_t^n, Y_t)$, on the right hand side in (25). In order for the estimation to work we also have to consider the second density, $p(X_t^n|Y_t)$, in (25). This can be written as

$$p(X_t^n|Y_t) = \frac{p(y_t|X_t^n, Y_{t-1})p(x_t^n|X_{t-1}^n, Y_{t-1})}{p(y_t|Y_{t-1})} p(X_{t-1}^n|Y_{t-1}), \tag{30}$$

where an approximation of $p(X_{t-1}^n|Y_{t-1})$ is provided by the previous iteration of the particle filter. Furthermore, we need analytical expressions for $p(y_t|X_t^n, Y_{t-1})$ and $p(x_t^n|X_{t-1}^n, Y_{t-1})$. They are provided by the following theorem.

Theorem C.2
For Model 4 we have that

$$p(y_t|X_t^n, Y_{t-1}) = \mathcal{N}(h_t + C_t\hat{x}_{t|t-1}^l,\, C_t P_{t|t-1} C_t^T + R_t), \tag{31a}$$

$$p(x_{t+1}^n|X_t^n, Y_t) = \mathcal{N}(f_t^n + A_t^n\hat{x}_{t|t}^l,\, A_t^n P_{t|t}(A_t^n)^T + G_t^n Q_t^n (G_t^n)^T). \tag{31b}$$

Proof See Appendix D. □

We are now ready to state the combined particle and Kalman filtering algorithm for estimating the state in the general model class (23).

Algorithm C.3 (The marginalized particle filter for Model 4)

1. *Initialization: For $i = 1, \ldots, N$, initialize the particles, $x_{0|-1}^{n,(i)} \sim p_{x_0^n}(x_0^n)$ and set $\{x_{0|-1}^{l,(i)}, P_{0|-1}^{(i)}\} = \{\bar{x}_0^l, \bar{P}_0\}$.*
2. *For $i = 1, \ldots, N$, evaluate the importance weights $q_t^{(i)} = p(y_t|X_t^{n,(i)}, Y_{t-1})$ according to (31a) and normalize $\tilde{q}_t^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^N q_t^{(j)}}$.*
3. *Particle filter measurement update: Resample with replacement N particles according to,*
$$Pr(x_{t|t}^{n,(i)} = x_{t|t-1}^{n,(j)}) = \tilde{q}_t^{(j)}.$$
4. *Particle filter time update and Kalman filter*
   (a) *Kalman filter measurement update, using (27).*
   (b) *Particle filter time update: For $i = 1, \ldots, N$, predict new particles using (31b) according to*
   $$x_{t+1|t}^{n,(i)} \sim p(x_{t+1|t}^n|X_t^{n,(i)}, Y_t).$$
   (c) *Kalman filter time update, using (28).*

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

As pointed out before, the only thing that makes this filter different from the standard particle filter is that the prediction stage is different here. To be more concrete, if steps 4*a* and 4*c* are removed in Algorithm C.3 we obtain Algorithm C.1.

6 Important Special Cases and Modeling Issues

There are certain versions of the general model class (23) which are more common and important than others, and this will, together with some modelling issues, be the topic for this section. This will be straightforward reductions of the general results stated in the previous section, however they still deserve some attention. One very important model class in applications is the one where the nonlinearity enters the model in the measurement equation and the state dynamics is linear. We will briefly discuss this class in the subsequent section, for a more thorough discussion, see Gustafsson *et al.* (2002).

We will now state some useful observations.

1. If there are no linear state variables, x_t^l, in the measurement equation (23c), i.e., $C_t = 0$, the density function of the measurement noise, e_t can be arbitrary, but it has to be known. The reason is that (23c) will then not contain any information about the linear variables, and hence it cannot be used in the Kalman filter, it is solely used in the particle filter, which can handle all densities.

2. Similarly, if $A_t^n = 0$ in (23a) this equation will be independent of the linear states, and hence it can not be used in the Kalman filter, which means that the state noise, w_t^n can be arbitrary, but it has to be known.

3. Another very important special case occurs when the matrices $A_t^n, A_t^l, G_t^n, G_t^l$ and C_t are independent of x_t^n. In this case we have that

$$P_{t|t}^{(i)} = P_{t|t} \qquad \forall i = 1, \ldots, N. \tag{32}$$

   This follows from (27b) – (27d) in Theorem C.1. When the conditions mentioned above are met, (32) will lead to that we only have to solve one instead of N Riccati equations, which leads to a substantial reduction in the computational load.

4. In this article we have used the most basic form of the particle filter. Several more refined variants exist, which can give better performance. However, since the aim of this article is to communicate the idea of marginalization in a general state-space model we have used the standard particle filter, as it was first introduced in Gordon *et al.* (1993). It is straightforward to adjust the Algorithms C.2 and C.3 to accommodate new ideas, such as e.g., the auxiliary particle filter, introduced in Pitt and Shephard (1999).

5. The noise covariances can depend on the nonlinear state variable, i.e., $R_t = R_t(x_t^n)$ and $Q_t = Q_t(x_t^n)$. This can be useful for instance in terrain navigation, as will be described in Section 7.

Having observed these important special cases we will now in the subsequent two section discuss some modelling issues relevant for the marginalized particle filter.

6.1 An Important Model Class

In this section we will study the following model class,

$$x_{t+1}^n = A_{n,t}^n x_t^n + A_{l,t}^n x_t^l + G_t^n w_t^n, \tag{33a}$$

$$x_{t+1}^l = A_{n,t}^l x_t^n + A_{l,t}^l x_t^l + G_t^l w_t^l, \tag{33b}$$

$$y_t = h_t(x_t^n) + e_t, \tag{33c}$$

which clearly is a special case of the general model class (23), corresponding to linear dynamics and a nonlinear measurement equation. The motivation for giving this model class special attention is that it is important in applications, e.g., positioning, target tracking and collision avoidance (Gustafsson *et al.*, 2002). Many important state estimation problems fit the model class (33). Usually the nonlinear state variable, x_t^n is the position, while the linear state variable, x_t^l, corresponds to the velocity and the acceleration. If we have an almost linear dynamics in (33a) – (33b) we can linearize it and use the extended Kalman filter instead of the Kalman filter. As is explained in Li and Jilkov (2000, 2001) it is common that the system model is almost linear, whereas the measurement model is severely nonlinear. In these cases it can be motivated to use the particle filter together with the extended Kalman filter.

When this model class is used the measurement equation (33c) does not contain any information about the linear state variable, x_t^l, and hence it is without information as far as the Kalman filter is concerned. Instead all the measurement information enters the Kalman filter implicitly via the artifical measurement given by the nonlinear state equation (33a). This means that in Algorithm C.3, step 4*a* can be left out. In this case the artifical measurement equation is much more than a correction of the real measurement, it is the only measurement information available. It is also worth noting that when model class (33) is used all the matrices are independent of the nonlinear variable, x_t^n, and hence we only need to solve one Riccati equation at each time step, according to what was said in the third observation in the previous section.

For more information on practical matters concerning modelling issues relevant in this respect, see e.g., (Gustafsson *et al.*, 2002; Li and Jilkov, 2000, 2001; Nordlund, 2002).

6.2 Augmenting the State Model

Linear subdynamics can enter the model more implicitly as well, e.g., via coloured state noise, sensor offsets and trends. This will be briefly sketched here. See

(Gustafsson, 2000, Section 8.2.4) for more on these topics.

Coloured State Noise

Let the original model be given by

$$x_{t+1}^n = f(x_t^n) + Bv_t, \tag{34a}$$
$$y_t = h(x_t^n) + e_t, \tag{34b}$$

where the noise, v_t, is coloured and can be modeled as

$$x_{t+1}^l = A^l x_t^l + B^l w_t^l, \tag{35a}$$
$$v_t = C^l x_t^l. \tag{35b}$$

The noise, w_t^l, is a sequence of independent, zero mean, Gaussian noise. We can now write the augmented system as

$$x_{t+1}^n = f(x_t^n) + BC^l x_t^l, \tag{36a}$$
$$x_{t+1}^l = A^l x_t^l + B^l w_t^l, \tag{36b}$$
$$y_t = h(x_t^n) + e_t, \tag{36c}$$

which is a special case of the general model class (23). For instance, in the tracking-community a common assumption is to assume that v_t is a first order Markov process as is done in the Singer model (Singer, 1970).

Sensor Offsets and Trends

Again the original model is given by (34a). Models for the slowly drifting sensor offsets are given by

$$x_{t+1}^l = Ix_t^l + S_w w_t^l, \tag{37}$$

where $w_t^l \in \mathcal{N}(0, I)$. The augmented system can now be written as

$$x_{t+1}^n = f(x_t^n) + Bv_t, \tag{38a}$$
$$x_{t+1}^l = Ix_t^l + S_w w_t^l, \tag{38b}$$
$$y_t = h(x_t^n) + Ix_t^l + e_t, \tag{38c}$$

It is straightforward to also include trends in a similar way.

7 Integrated Aircraft Navigation

In this section the theory discussed above will be used in an aircraft navigation system. The purpose of a navigation system is to obtain an estimate of the aircrafts position, orientation and velocity. The core of most aircraft navigation systems of

today is an Inertial Navigation System (INS), which uses on-board acceleration and angular velocity sensors. The data from these sensors are integrated in order to obtain the position, velocity and heading. The problem with using the INS is that the estimate will deteriorate with time, and hence we need more information to base our estimates on. When the INS is complemented with one or more additional sensors we obtain an integrated navigation system. Common examples of additional sensors are the Global Positioning System (GPS) or Terrain Aided Positioning (TAP). In TAP, a terrain elevation database together with height measurements is used to obtain an estimate of the position. It is the highly nonlinear nature of the terrain elevation database, together with the non-Gaussian measurement noise that motivates the use of the particle filter in this application. See e.g., Bergman *et al.* (1999) for an introduction to aircraft navigation in general and terrain navigation in particular.

7.1 The Dynamic Model

In order to apply the marginalized particle filter to the navigation problem we need a dynamic model of the aircraft. We will in this section only discuss the structure of this model, for details the reader is referred to Nordlund (2002). Due to the often very fast dynamics of an aircraft we will estimate the errors of the states instead of the states themselves. This will provide better estimates, since the dynamics of the errors is typically much slower compared to the actual states. The model has the following structure

$$x^n_{t+1} = A^n_{n,t}x^n_t + A^n_{l,t}x^l_t + G^n_t w^n_t, \tag{39a}$$

$$x^l_{t+1} = A^l_{n,t}x^n_t + A^l_{l,t}x^l_t + G^l_t w^l_t, \tag{39b}$$

$$y_t = h\left(\begin{bmatrix} L_t \\ l_t \end{bmatrix} + x^n_t\right) + e_t. \tag{39c}$$

There are 7 linear states, and 2 nonlinear states. The linear states consist of 2 velocity states, 3 states for the aircraft in terms of heading, roll, and pitch. Finally, there are 2 states for the accelerometer bias. The two nonlinear states correspond to the horisontal position expressed in latitude, L_t and longitude, l_t.

The total dimension of the state vector is thus 9, which can be hard for the particle filter to handle, since it would require a large number of particles. This would in turn imply large computational demands. Furthermore, the highly nonlinear nature of measurement equation (39c), due to the terrain elevation database, implies that we cannot use an extended Kalman filter. However the model class (39) clearly fits into the framework of the marginalized particle filter, compare with (23).

The measurement noise in (39c) deserves some special attention. The radar altimeter, which is used to measure the ground clearance, interprets any echo as the ground. This is a problem when flying over trees, since the tree tops will then also be interpreted as the ground, with a false measurement as a result. One solution to this problem is to model the measurement noise as

$$p_{e_t} = \pi\mathcal{N}(m_1, \sigma_1) + (1-\pi)\mathcal{N}(m_2, \sigma_2), \tag{40}$$

where π is the probability of obtaining an echo from the ground, and $(1-\pi)$ is the probability of obtaining en echo from the tree tops. The density (40) is also shown in Figure 1. Empirical experiments at Saab Aerospace have shown that

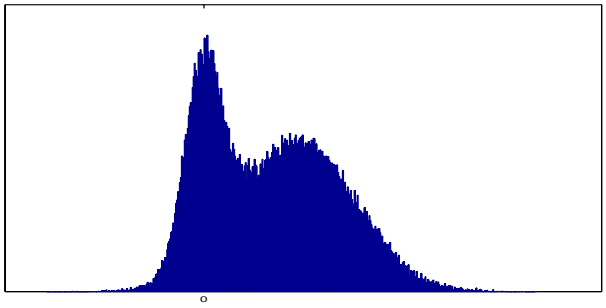


Figure 1 *A typical histogram of the error in the data from the radar altimeter. The first top corresponds to the error in the ground reading and the second top corresponds to the error in the readings from the tree tops.*

this, in spite of its simplicity, is a quite accurate model. Note that in (40) we can have $p_{e_t} = p_{e_t}(x_t^n)$, that is m_1, m_2, σ_1, and σ_2 depend on the current position. In this way we can infer information from the terrain data base on the measurement model.

7.2 Result

A navigation system based on the marginalized particle filter has been tested on *authentic flight data* recorded during real flights with the Swedish fighter aircraft JAS 39 Gripen. In this section we present the results. For a more complete discussion of the results see Frykman (2003). The flight that has been used is shown in Figure 2. This is a fairly tough flight for the algorithm, in the sense that during some intervals data are missing, and sometimes the radar altimeter readings become unreliable. This happens at high altitudes and during sharp turns respectively. In order to get a fair understanding of the algorithms performance, 100 Monte Carlo simulations of the same data have been used, where only the noise realizations have been changed from one simulation to the other. There are a lot of parameters that have to be chosen, here we will only comment on the number of particles, for more details see Frykman (2003). In Figure 3 below we present a plot of the horisontal position (L_t, l_t) error as a function time, for different number of particles. From this figure it is obvious that the estimate is better the more particles we use, which is natural since the more particles we use the more accurately the involved densities are approximated. We also see that the difference in performance is mainly during the transient, where it can be motivated to use more particles, hence we choose to

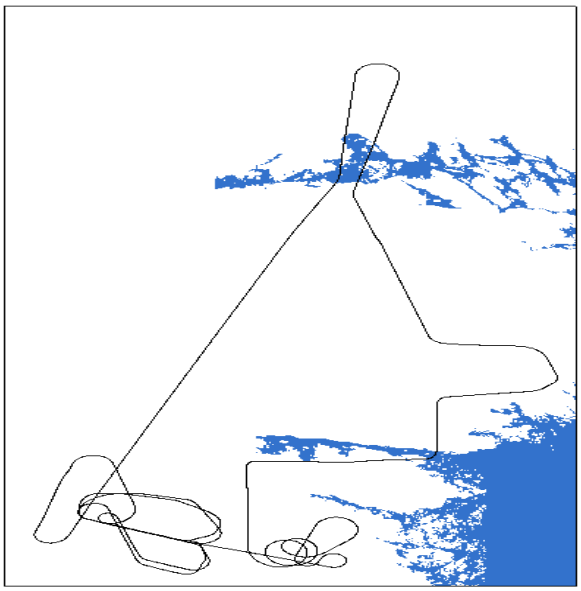


Figure 2 *The flight path used for testing the algorithm. The flight path is clockwise and the dark regions in the figure are the lake Mälaren and the Baltic sea.*

use 5000 particles for this study.

In Figure 4 the estimation error in the horisontal position is shown, together with the altitude profile of the aircraft and the ground elevation. The true position is provided by the differential GPS (DGPS). During two intervals (illustrated in the upper plot in Figure 4), when the aircraft is flying at a very high altitude the radar altimeter does not deliver any information. From the bottom plot in Figure 4 we see that the estimation error will grow during this intervals. However, when the measurements return the estimate converges again. Towards the end of the flight the estimation error grows, due to the sharp turns (see Figure 2). The reason is that there is not enough time for the algorithm to converge between the turns. The algorithm can be further improved, several suggestions are given in Frykman (2003).

The conclusion from this study is that the marginalized particle filter performs well, and it provides an interesting and powerful alternative to the methods currently used in integrated aircraft navigation systems.

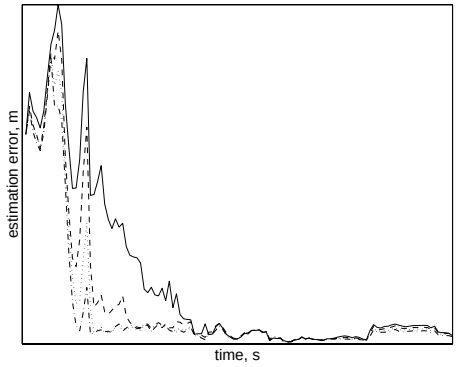


Figure 3 *The horisontal position (L_t, l_t) error as a function of the number of particles. The solid line corresponds to* 1200 *particles, the dashed* 2500 *particles, the dotted* 5000 *particles, and the dash-dotted* 10000 *particles. We have used the marginalized particle filter given in Algorithm C.3.*

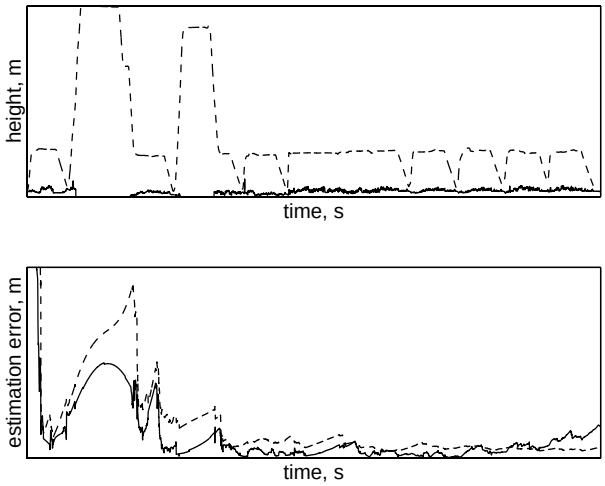


Figure 4 *In the top plot the altitude profil of the aircraft (dashed) and the ground elevation (solid) is shown. The bottom plot shows the horisontal estimation error (solid) and the corresponding standard deviation (dashed).*

8 Conclusions

We have systematically applied marginalization techniques to nonlinear and non-Gaussian state-space models, where certain states appear linearly in the model.

We have done this in several steps, where each step adds a certain modification to a generic particle filter implementation. The first step is to attach a Kalman filter to each particle, so that each particle represents the nonlinear state and the conditional mean and covariance, respectively, of the linear states. The second main step implies that the state prediction in the particle filter must be seen as an artificial measurement for the linear state. This implied a further measurement update in the Kalman filter, and thus one additional step in the modified particle filter.

We also along the road pointed out several important special cases, for instance conditions for all Ricatti equations of the Kalman filter to be the same and how to linearize almost linear states so the Kalman filters are replaced by extended Kalman filters to mention a few. It is also described how colored noise, offsets and trends automatically leads to linear sub-structures that can be exploited by this approach.

Finally, a terrain navigation application with real data from the fighter JAS 39 Gripen was presented. The particle filter is not a feasible algorithm for the full nine-state model because of that a huge number of particles would be needed. However, since only two states (the aircraft's horizontal position) appear nonlinearly in the measurement equation, a special case of the general marginalization algorithm can be applied, and a very good result can be obtained with only 5000 particles, which readily is possible to implement in the existing aircraft computer.

Appendix

In the appendices below we provide the proofs of the lemmas and the theorems stated in this article.

A Proof for Lemma C.1

Proof According to (8a) the nonlinear states at the next time instant, x^n_{t+1}, are independent of the linear states at the current time instant, x^l_t. Put in other words (8a) does not contain any information about the linear states. This implies that if we assume that x^n_t and y_t are known, the model as far as the linear states are concerned, is given by

$$x^l_{t+1} = A^l_t x^l_t + w^l_t, \tag{41a}$$

$$y_t = h_t + C_t x^l_t + e_t, \tag{41b}$$

where we have suppressed the fact that the matrices A^l_t, C_t, and h_t are dependent of x^n_t. Since we have assumed that x^n_t is known the matrices A^l_t, C_t, and h_t are constant matrices at time t. Model (41) is linear and Gaussian, and hence the optimal estimate of the linear state, x^l_t, is given by the Kalman filter (Anderson

and Moore, 1979) according to

$$\hat{x}^l_{t|t} = \hat{x}^l_{t|t-1} + K_t(y_t - h_t - C_t\hat{x}^l_{t|t-1}), \tag{42a}$$
$$P_{t|t} = P_{t|t-1} - K_t C_t P_{t|t-1}, \tag{42b}$$
$$S_t = C_t P_{t|t-1} C_t^T + R_t, \tag{42c}$$
$$K_t = P_{t|t-1} C_t^T S_t^{-1}, \tag{42d}$$
$$\hat{x}^l_{t+1|t} = A^l_t \hat{x}^l_{t|t}, \tag{42e}$$
$$P_{t+1|t} = A^l_t P_{t|t} (A^l_t)^T + Q^l_t. \tag{42f}$$

□

B Proof for Lemma C.2

Proof We start by writing $p(y_t, x^l_t|X^n_t, Y_{t-1})$ according to

$$\begin{aligned} p(y_t, x^l_t|X^n_t, Y_{t-1}) &= p(y_t|x^l_t, X^n_t, Y_{t-1}) p(x^l_t|X^n_t, Y_{t-1}) \\ &= p(y_t|x^l_t, x^n_t) p(x^l_t|X^n_t, Y_{t-1}). \end{aligned} \tag{43}$$

The second equality above follows from the fact that if we know x^l_t and x^n_t there will not be any additional information in X^n_{t-1} and Y_{t-1}. Now we have

$$\begin{aligned} p(y_t|X^n_t, Y_{t-1}) &= \int p(y_t, x_t|X^n_t, Y_{t-1}) dx^l_t \\ &= \int p(y_t|x^l_t, x^n_t) p(x^l_t|X^n_t, Y_{t-1}) dx^l_t \end{aligned} \tag{44}$$

where

$$p(y_t|x^l_t, x^n_t) = \mathcal{N}(h_t(x^n_t) + C_t(x^n_t)x^l_t, R_t), \tag{45}$$

according to (8c). From Lemma C.1 we have

$$p(x^l_t|X^n_t, Y_{t-1}) = \mathcal{N}(\hat{x}^l_{t|t-1}, P_{t|t-1}). \tag{46}$$

Writing out (45) and (46) explicitly gives (we will, for the sake of brevity, from now on suppress the dependence on x^n_t)

$$p(y_t|x^l_t, x^n_t) = \frac{1}{(2\pi)^{\frac{n_y}{2}} (\det[R_t])^{\frac{1}{2}}} e^{-\frac{1}{2}(y_t - h_t - C_t x^l_t) R_t^{-1} (y_t - h_t - C_t x^l_t)} \tag{47a}$$

$$p(x^l_t|X^n_t, Y_{t-1}) = \frac{1}{(2\pi)^{\frac{n_{x^l}}{2}} (\det\left[P_{t|t-1}\right])^{\frac{1}{2}}} e^{-\frac{1}{2}(x^l_t - \hat{x}^l_{t|t-1})^T P^{-1}_{t|t-1} (x^l_t - \hat{x}^l_{t|t-1})} \tag{47b}$$

Inserting (47a) and (47b) in (44) gives

$$\begin{aligned} p(y_t|X^n_t, Y_{t-1}) = \int \frac{1}{(2\pi)^{\frac{n_{x^l} + n_y}{2}} (\det[R_t] \det\left[P_{t|t-1}\right])^{\frac{1}{2}}} \cdot \\ e^{-\frac{1}{2}(\epsilon_t - C_t \tilde{x}^l_{t|t-1})^T R_t^{-1} (\epsilon_t - C_t \tilde{x}^l_{t|t-1}) - \frac{1}{2}(\tilde{x}^l_{t|t-1})^T P^{-1}_{t|t-1} (\tilde{x}^l_{t|t-1})} dx^l_t, \end{aligned} \tag{48}$$

where we have introduced

$$\tilde{x}^l_{t|t-1} = x^l_t - \hat{x}^l_{t|t-1}, \tag{49a}$$

$$\epsilon_t = y_t - h_t - C_t \hat{x}^l_{t|t-1}. \tag{49b}$$

The exponent in (48) can be written as

$$\begin{bmatrix} \tilde{x}^l_{t|t-1} \\ \epsilon_t \end{bmatrix}^T \underbrace{\begin{bmatrix} (P^{-1}_{t|t-1} + C_t^T R_t^{-1} C_t) & -C_t^T R_t^{-1} \\ -R_t^{-1} C_t & R_t^{-1} \end{bmatrix}}_{H} \begin{bmatrix} \tilde{x}^l_{t|t-1} \\ \epsilon_t \end{bmatrix}. \tag{50}$$

The matrix H can be factored according to

$$H = \begin{bmatrix} I & -K_t \\ 0 & I \end{bmatrix}^T \begin{bmatrix} P^{-1}_{t|t} & 0 \\ 0 & S_t^{-1} \end{bmatrix} \begin{bmatrix} I & -K_t \\ 0 & I \end{bmatrix}, \tag{51}$$

where

$$K_t = (P^{-1}_{t|t-1} + C_t^T R_t^{-1} C_t)^{-1} C_t^T R_t^{-1}, \tag{52a}$$

$$P^{-1}_{t|t} = C_t^T R_t^{-1} C_t + P^{-1}_{t|t-1}, \tag{52b}$$

$$S_t^{-1} = R_t^{-1} - R_t^{-1} C_t (C_t^T R_t^{-1} C_t + P^{-1}_{t|t-1})^{-1} C_t^T R_t^{-1}. \tag{52c}$$

Using the matrix inversion lemma

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1} \tag{53}$$

we can rewrite (52a) according to

$$K_t = (P_{t|t-1} - P_{t|t-1} C_t^T (C_t P_{t|t-1} C_t^T + R_t)^{-1} C_t P_{t|t-1}) C_t^T R_t^{-1}. \tag{54}$$

If we continue rewriting (54) we obtain

$$\begin{aligned} K_t &= P_{t|t-1} C_t^T (I - (C_t P_{t|t-1} C_t^T + R_t)^{-1} C_t P_{t|t-1} C_t^T) R_t^{-1} \\ &= P_{t|t-1} C_t^T ((C_t P_{t|t-1} C_t^T + R_t)^{-1} (C_t P_{t|t-1} C_t^T + R_t - C_t P_{t|t-1} C_t^T) R_t^{-1} \\ &= P_{t|t-1} C_t^T (C_t P_{t|t-1} C_t^T + R_t)^{-1} R_t R_t^{-1} \\ &= P_{t|t-1} C_t^T (C_t P_{t|t-1} C_t^T + R_t)^{-1}. \end{aligned} \tag{55}$$

Direct application of the matrix inversion lemma (53) to (52b) and (52c) gives

$$P_{t|t} = P_{t|t-1} + P_{t|t-1} C_t^T (C_t P_{t|t-1} C_t^T + R_t)^{-1} C_t P_{t|t-1}, \tag{56a}$$

$$S_t = C_t P_{t|t-1} C_t^T + R_t. \tag{56b}$$

Inserting (51) in (50) gives

$$\begin{aligned} &\begin{bmatrix} \tilde{x}^l_{t|t-1} - K_t \epsilon_t \\ \epsilon_t \end{bmatrix}^T \begin{bmatrix} P^{-1}_{t|t} & 0 \\ 0 & S_t^{-1} \end{bmatrix} \begin{bmatrix} \tilde{x}^l_{t|t-1} - K_t \epsilon_t \\ \epsilon_t \end{bmatrix} \\ &= (\tilde{x}^l_{t|t-1} - K_t \epsilon_t)^T P^{-1}_{t|t} (\tilde{x}^l_{t|t-1} - K_t \epsilon_t) + \epsilon_t^T S_t^{-1} \epsilon_t \end{aligned} \tag{57}$$

Let us now rewrite the determinant in (48) according to

$$\begin{aligned}\frac{1}{\det\left[R_t\right]\det\left[P_{t|t-1}\right]} &= \det\left[R_t^{-1}\right]\det\left[P_{t|t-1}^{-1}\right] = \det\begin{bmatrix}P_{t|t-1}^{-1} & 0\\ 0 & R_t^{-1}\end{bmatrix}\\ &= \det\left[\begin{bmatrix}I & K_t\\ 0 & I\end{bmatrix}^T\begin{bmatrix}I & 0\\ -C_t & I\end{bmatrix}^T\begin{bmatrix}P_{t|t-1}^{-1} & 0\\ 0 & R_t^{-1}\end{bmatrix}\begin{bmatrix}I & 0\\ -C_t & I\end{bmatrix}\begin{bmatrix}I & K_t\\ 0 & I\end{bmatrix}\right]\\ &= \det\begin{bmatrix}P_{t|t}^{-1} & 0\\ 0 & S_t^{-1}\end{bmatrix} = \det\left[P_{t|t}^{-1}\right]\det\left[S_t^{-1}\right] = \frac{1}{\det\left[P_{t|t}\right]\det\left[S_t\right]}.\end{aligned} \tag{58}$$

In the third equality above we have used the fact that the determinant of a triangular matrix with ones in the diagonal equals one. Using (57) and (58) in (48) gives

$$\begin{aligned}p(y_t|X_t^n, Y_{t-1}) = \int &\frac{1}{(2\pi)^{\frac{n_{x^l}}{2}}(\det\left[P_{t|t}\right])^{\frac{1}{2}}}e^{-\frac{1}{2}(\tilde{x}_{t|t-1}^l - K_t\epsilon_t)^T P_{t|t}^{-1}(\tilde{x}_{t|t-1}^l - K_t\epsilon_t)}dx_t^l\cdot\\ &\frac{1}{(2\pi)^{\frac{n_y}{2}}(\det\left[S_t\right])^{\frac{1}{2}}}e^{-\frac{1}{2}\epsilon_t^T S_t^{-1}\epsilon_t} = \frac{1}{(2\pi)^{\frac{n_y}{2}}(\det\left[S_t\right])^{\frac{1}{2}}}e^{-\frac{1}{2}\epsilon_t^T S_t^{-1}\epsilon_t}\end{aligned} \tag{59}$$

In the last equality above we have used the fact that the integral of a probability density function over its entire range equals one. We have now proved that

$$p(y_t|X_t^n, Y_{t-1}) = \mathcal{N}(h_t + C_t\hat{x}_{t|t-1}^l,\, C_t P_{t|t-1}C_t^T + R_t) \tag{60}$$

The density $p(x_{t+1}^n|X_t^n, Y_t)$ can analogously to (44) be written as

$$p(x_{t+1}^n|X_t^n, Y_t) = \int p(x_{t+1}^n|x_t^l, x_t^n)p(x_t^l|X_t^n, Y_t)dx_t^l \tag{61}$$

where

$$p(x_{t+1}^n|x_t^l, x_t^n) = \mathcal{N}(f_t^n, Q_t^n) \tag{62}$$

according to (8a) and

$$p(x_t^l|X_t^n, Y_t) = \mathcal{N}(\hat{x}_{t|t}^l, P_{t|t}) \tag{63}$$

according to the result in Lemma C.1. Now, performing the integration in (61) using the two densities (62) and (63) proves the second part of the lemma. □

C Proof for Theorem C.1

The proof of (21) and (22) is provided as a special case of the proof below.

Proof For the sake of brevity we will suppress the dependence on x_t^n in (23) in this proof. Let us start by writing (23) as

$$x_{t+1}^l = f_t^l + A_t^l x_t^l + G_t^l w_t^l, \tag{64a}$$
$$z_t^1 = A_t^n x_t^l + G_t^n w_t^n, \tag{64b}$$
$$z_t^2 = C_t x_t^l + e_t, \tag{64c}$$

where z_t^1 and z_t^2 are defined as

$$z_t^1 = x_{t+1}^n - f_t^n, \tag{64d}$$
$$z_t^2 = y_t - h_t, \tag{64e}$$

Inspection of the above equations gives that z_t^1 and z_t^2 can both be thought of as measurements, since mathematically (64b) and (64c) possess the structure of measurement equations. Before we can go on we have to take care of the fact that there is a cross-correlation between the two noise processes w_t^l and w_t^n, since $Q_t^{ln} \neq 0$ in (24a). We can use the Gram-Schmidt procedure to de-correlate the noise (Gustafsson, 2000; Kailath *et al.*, 2000). Instead of w_t^l we can use

$$\begin{aligned}\bar{w}_t^l &= w_t^l - \mathrm{E}[w_t^l (w_t^n)^T](\mathrm{E}[w_t^n (w_t^n)^T])^{-1} w_t^n \\ &= w_t^l - Q_t^{ln}(Q_t^n)^{-1} w_t^n = w_t^l - Q_t^{ln}(Q_t^n)^{-1}(G_t^n)^{-1}(z_t^1 - A_t^n x_t^l),\end{aligned} \tag{65}$$

resulting in $\mathrm{E}[\bar{w}_t^l (w_t^n)^T] = 0$ and

$$\bar{Q}_t^l = \mathrm{E}[\bar{w}_t^l (\bar{w}_t^l)^T] = Q_t^l - Q_t^{ln}(Q_t^n)^{-1} Q_t^{ln}. \tag{66}$$

We can now rewrite (64a) using (64b) and (65) according to (we assume that G_t^n is invertible. The case of a noninvertible G_t^n is treated in Bergman (1999))

$$\begin{aligned}x_{t+1}^l &= A_t^l x_t^l + G_t^l[\bar{w}_t^l + Q_t^{ln}(Q_t^n)^{-1}(G_t^n)^{-1}(z_t^1 - A_t^n x_t^l)] + f_t^l, \\ &= \bar{A}_t^l x_t^l + G_t^l \bar{w}_t^l + G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1} z_t^1 + f_t^l,\end{aligned} \tag{67}$$

where

$$\bar{A}_t^l = A_t^l - G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1} A_t^n. \tag{68}$$

We can now write our de-correlated system as

$$x_{t+1}^l = f_t^l + \bar{A}_t^l x_t^l + G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1} z_t^1 + G_t^l \bar{w}_t^l, \tag{69a}$$
$$z_t^1 = A_t^n x_t^l + G_t^n w_t^n, \tag{69b}$$
$$z_t^2 = C_t x_t^l + e_t, \tag{69c}$$

which is a linear system with Gaussian noise. Moreover, from (64d) and (64e) we have that Z_t^1 and Z_t^2 are known if X_{t+1}^n and Y_t are known. We are now ready to start the actual proof of the theorem, which will be done using induction. At time

zero we have that $p(x_0^l|X_0^n, Y_{-1}) = p(x_0^l|x_0^n) = \mathcal{N}(\bar{x}_0^l, \bar{P}_0)$. Let us now assume that $p(x_t^l|X_t^n, Y_{t-1})$ is Gaussian at an arbitrary time, t.

The recursions are now divided into three parts. The first part consists of using the information available in the actual measurement, y_t, i.e., z_t^2. Once this measurement update has been done we will have the estimate, $\hat{x}_{t|t}^l$ and $P_{t|t}$. These can now be used to calculate the predictions of the nonlinear state, $x_{t+1|t}^n$. These prediction will provide us with new information about the system and hence the second part is to incorporate this new information by performing a measurement update using the artificial measurement, z_t^1. Finally, the third part consists of a time update using the result from the second step.

Part 1: Assume that z_t^2 is available.
(i) According to the initial assumptions we have $p(x_0^l|X_0^n, Y_{t-1}) = p(x_0^l|x_0^n) = \mathcal{N}(\bar{x}_0, \bar{P}_0)$.
(ii) Assume that $p(x_t^l|X_t^n, Y_{t-1}) = \mathcal{N}(\hat{x}_{t|t-1}^l, P_{t|t-1})$. We now want to incorporate a new measurement, y_t. This can be done by computing

$$p(x_t^l|X_t^n, Y_t) = \frac{p(y_t|x_t^n, x_t^l)p(x_t^l|X_t^n, Y_{t-1})}{\int p(y_t|x_t^n, x_t^l)p(x_t^l|X_t^n, Y_{t-1})dx_t^l}. \tag{70}$$

Using the fact that the measurement noise and thereby $p(y_t|x_t^n, x_t^l)$ is Gaussian, i.e.,

$$p(y_t|x_t^n, x_t^l) = \frac{1}{(2\pi)^{\frac{n_y}{2}}(\det[R_t])^{\frac{1}{2}}} e^{-\frac{1}{2}(y_t - h_t - C_t x_t^l)^T R_{t|t}^{-1}(y_t - h_t - C_t x_t^l)}. \tag{71}$$

According to the induction assumption we have

$$p(x_t^l|X_t^n, Y_{t-1}) = \frac{1}{(2\pi)^{\frac{n_{x^l}}{2}}(\det\left[P_{t|t-1}\right])^{\frac{1}{2}}} e^{-\frac{1}{2}(x_t^l - \hat{x}_{t|t-1}^l)^T P_{t|t-1}^{-1}(x_t^l - \hat{x}_{t|t-1}^l)} \tag{72}$$

Now, the denominator in (70) is exactly (48), hence we have

$$p(y_t|X_t^n, Y_{t-1}) = \frac{1}{(2\pi)^{\frac{n_y}{2}}(\det[S_t])^{\frac{1}{2}}} e^{-\frac{1}{2}\epsilon_t^T S_t^{-1}\epsilon_t}, \tag{73}$$

where $\epsilon_t = y_t - h_t - C_t\hat{x}_{t|t-1}^l$ and $S_t = C_t P_{t|t-1} C_t^T$. In order to calculate the numerator in (70), we can use (59) and (73) resulting in

$$p(x_t^l|X_t^n, Y_t) = \frac{1}{(2\pi)^{\frac{n_{x^l}}{2}}(\det\left[P_{t|t}\right])^{\frac{1}{2}}} e^{-\frac{1}{2}(x_t^l - \hat{x}_{t|t}^l)^T P_{t|t}^{-1}(x_t^l - \hat{x}_{t|t}^l)}. \tag{74}$$

Hence, we have $p(x_t^l|X_t^n, Y_t) = \mathcal{N}(\hat{x}_{t|t}^l, P_{t|t})$ where

$$\hat{x}_{t|t}^l = \hat{x}_{t|t-1}^l + K_t(y_t - h_t - C_t\hat{x}_{t|t-1}^l), \tag{75a}$$
$$P_{t|t} = P_{t|t-1} - K_t M_t K_t^T, \tag{75b}$$
$$K_t = P_{t|t-1} C_t^T M_t^{-1}, \tag{75c}$$
$$M_t = C_t P_{t|t-1} C_t^T + R_t. \tag{75d}$$

(iii) According to (i), (ii), and the principle of induction we have now proved the first part.

Part 2: At this stage z_t^1 becomes available. Now using

$$p(x_t^l|X_{t+1}^n, Y_t) = \frac{p(x_{t+1}^n|x_t^n, x_t^l)p(x_t^l|X_t^n, Y_t)}{\int p(x_{t+1}^n|x_t^n, x_t^l)p(x_t^l|X_t^n, Y_t)dx_t^l} \tag{76}$$

analogously to part 1 we obtain

$$\hat{x}_{t|t}^{l*} = \hat{x}_{t|t}^l + L_t(z_t^1 - A_t^n \hat{x}_{t|t}^l), \tag{77a}$$
$$P_{t|t}^* = P_{t|t} - L_t N_t^* L_t^T, \tag{77b}$$
$$L_t = P_{t|t}(A_t^n)^T (N_t^*)^{-1}, \tag{77c}$$
$$N_t^* = A_t^n P_{t|t}(A_t^n)^T + G_t^n Q_t^n (G_t^n)^T. \tag{77d}$$

Part 3: The final part of this proof is the time update, i.e., to compute

$$p(x_{t+1}^l|X_{t+1}^n, Y_t) = \int p(x_{t+1}^l|x_{t+1}^n, x_t^n, x_t^l)p(x_t^l|X_{t+1}^n, Y_t)dx_t^l. \tag{78}$$

According to part 2 above we have that

$$p(x_t^l|X_{t+1}^n, Y_t) = \frac{1}{(2\pi)^{\frac{n_{x^l}}{2}}(\det\left[P_{t|t}^*\right])^{\frac{1}{2}}} e^{-\frac{1}{2}(x_t^l - \hat{x}_{t|t}^{l*})^T (P_{t|t}^{-1})^{-1}(x_t^l - \hat{x}_{t|t}^{l*})} \tag{79}$$

Furthermore, if we assume that G_t^l is nonsingular we have

$$e^{-\frac{1}{2}(x_{t+1}^l - \bar{A}_t^l - G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1} z_t^1 - f_t^l)^T \tilde{Q}^{-1}(x_{t+1}^l - \bar{A}_t^l - G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1} z_t^1 - f_t^l)}.$$

where

$$\tilde{Q}_t = G_t^l Q_t^l (G_t^l)^T \tag{80}$$

The assumption that G_t^l is nonsingular is just for notational reasons. The case with a singular G_t^l can be solved using the singular value decomposition (SVD). For details regarding this matter the reader is referred to Nordlund (2002). Now, using (79) and (80) we have

$$p(x_{t+1}^l|X_{t+1}^n, Y_t) = \int \frac{1}{(2\pi)^{n_{x^l}}\left(\det\left[P_{t|t}^*\right]\det\left[\tilde{Q}_t\right]\right)^{\frac{1}{2}}} \cdot$$
$$e^{-\frac{1}{2}(\tilde{x}_{t|t}^l)^T (P_{t|t}^*)^{-1}\tilde{x}_{t|t}^l - \frac{1}{2}(\xi_{t+1} - \bar{A}_t^l \hat{x}_{t|t}^l)^T \tilde{Q}_t^{-1}(\xi_{t+1} - \bar{A}_t^l \hat{x}_{t|t}^l)} dx_t \tag{81}$$

where we have used

$$\tilde{x}_{t|t}^l = x_t^l - \hat{x}_{t|t}^{l*}, \tag{82}$$
$$\xi_{t+1} = x_{t+1}^l - \bar{A}_t^l - G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1} z_t^1 - f_t^l. \tag{83}$$

We can now rewrite the exponent in (81) analogously to what was done in the proof for Lemma C.2.

$$(\tilde{x}^l_{t|t})^T (P^*_{t|t})^{-1} \tilde{x}^l_{t|t} - (\xi_{t+1} - \bar{A}^l_t \hat{x}^l_{t|t})^T \tilde{Q}^{-1}_t (\xi_{t+1} - \bar{A}^l_t \hat{x}^l_{t|t}) \tag{84}$$

$$= \begin{bmatrix} \tilde{x}^l_{t|t} \\ \xi_{t+1} \end{bmatrix}^T \begin{bmatrix} I & -L_t \\ 0 & I \end{bmatrix}^T \begin{bmatrix} \Theta^{-1}_t & 0 \\ 0 & P^{-1}_{t+1|t} \end{bmatrix} \begin{bmatrix} I & -L_t \\ 0 & I \end{bmatrix} \begin{bmatrix} \tilde{x}^l_{t|t} \\ \xi_{t+1} \end{bmatrix} \tag{85}$$

where we have used

$$\Theta_t = P^*_{t|t} - P^*_{t|t} \bar{A}^T_t (\tilde{Q}_t + \bar{A}^l_t P^*_{t|t} \bar{A}^l_t)^{-1} \bar{A}^l_t P^*_{t|t}, \tag{86a}$$

$$P_{t+1|t} = \bar{A}^l_t P^*_{t|t} \bar{A}^l_t + \tilde{Q}_t, \tag{86b}$$

$$L_t = P^*_{t|t} \bar{A}^l_t (\bar{A}^l_t P^*_{t|t} \bar{A}^l_t + \tilde{Q}_t)^{-1} \tag{86c}$$

Now using the same technique as was used in (59) we obtain

$$p(x^l_{t+1}|X^n_{t+1}, Y_t) = \frac{1}{(2\pi)^{\frac{n_{x^l}}{2}} \left(\det\left[P_{t+1|t}\right]\right)^{\frac{1}{2}}} e^{-\frac{1}{2}\xi^T_{t+1} P^{-1}_{t+1|t} \xi_{t+1}} \tag{87}$$

We have now derived $p(x^l_{t+1}|X^n_{t+1}, Y_t) = \mathcal{N}(\hat{x}^l_{t+1|t}, P_{t+1|t})$ where

$$\begin{aligned} \hat{x}^l_{t+1|t} &= \bar{A}^l_t \hat{x}^{l*}_{t|t} + G^l_t Q^{ln}_t (G^n_t Q^n_t)^{-1} z^1_t + f^l_t \\ &= \bar{A}^l_t \hat{x}^l_{t|t} + G^l_t Q^{ln}_t (G^n_t Q^n_t)^{-1} z^1_t + f^l_t + L_t (z^1_t - A^n_t \hat{x}^l_{t|t}), \end{aligned} \tag{88a}$$

$$\begin{aligned} P_{t+1|t} &= \bar{A}^l_t P^*_{t|t} \bar{A}^l_t + \tilde{Q}_t = \bar{A}^l_t (P_{t|t} - L_t N^*_t L^T_t) \bar{A}^l_t + G^l_t \bar{Q}^l_t (G^l_t)^T \\ &= \bar{A}^l_t P_{t|t} (\bar{A}^l_t)^T + G^l_t \bar{Q}^l_t (G^l_t)^T - L_t N_t L^T_t. \end{aligned} \tag{88b}$$

Finally, we have arrived in that $p(x^l_{t+1}|X^n_{t+1}, Y_t) = \mathcal{N}(\hat{x}^l_{t+1|t}, P_{t+1|t})$ where

$$\hat{x}^l_{t+1|t} = \bar{A}^l_t \hat{x}^l_{t|t} + G^l_t Q^{ln}_t (G^n_t Q^n_t)^{-1} z^1_t + f^l_t + L_t (z^1_t - A^n_t \hat{x}^l_{t|t}), \tag{89a}$$

$$P_{t+1|t} = \bar{A}^l_t P_{t|t} (\bar{A}^l_t)^T + G^l_t \bar{Q}^l_t (G^l_t)^T - L_t N_t L^T_t, \tag{89b}$$

$$L_t = \bar{A}^l_t P_{t|t} (A^n_t)^T N^{-1}_t, \tag{89c}$$

$$N_t = A^n_t P_{t|t} (A^n_t)^T + G^n_t Q^n_t (G^n_t)^T. \tag{89d}$$

The proof is now complete. □

D Proof for Theorem C.2

Proof This proof is exactly the same as the proof for Lemma C.2 given in Appendix B, save the fact that in this more general case we have to use

$$p(x^n_{t+1}|x^l_t, x^n_t) = \mathcal{N}(f^n_t + A^l_t \hat{x}^l_{t|t}, G^n_t Q^n_t (G^n_t)^T), \tag{90}$$

to obtain the result. Assumption (24) gives (90). □

Acknowledgment

The authors would like to thank Petter Frykman for providing the plots in Section 7, which were generated as a part of his Master's thesis at Saab Aerospace in Linköping, Sweden.

References

Anderson, B. and Moore, J. (1979). *Optimal Filtering*. Information and system science series. Prentice Hall, Englewood Cliffs, New Jersey.

Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society*.

Bergman, N. (1999). *Recursive Bayesian Estimation: Navigation and Tracking Applications*. PhD thesis, Linköping University. Dissertation No. 579.

Bergman, N., Ljung, L., and Gustafsson, F. (1999). Terrain navigation using Bayesian statistics. *IEEE Control Systems Magazine*, 19(3):33–40.

Chen, R. and Liu, J. (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society*, 62(3):493–508.

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.

Doucet, A., Gordon, N., and Krishnamurthy, V. (1999). Particle filters for state estimation of jump Markov linear systems. Technical Report CUED/F-INFENG/TR 359, Signal Processing Group, Department of Engineering, University of Cambridge, Trupington street, CB2 1PZ Cambridge.

Doucet, A., Gordon, N., and Krishnamurthy, V. (2001b). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624.

Frykman, P. (2003). Applied particle filters in integrated aircraft navigation. LiTH-ISY-EX-3406, Automatic control and communications systems, Linköping university.

Gordon, N., Salmond, D., and Smith, A. (1993). A 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.

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.

Kailath, T., Sayed, A., and Hassibi, B. (2000). *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, New Jersey.

Li, X. and Jilkov, V. (2000). A survey of maneuvering target tracking: Dynamic models. In *proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, Orlando, USA.

Li, X. and Jilkov, V. (2001). A survey of maneuvering target tracking - part iii: Measurement models. In *proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, San Diego, USA.

Nordlund, P.-J. (2002). Sequential Monte Carlo filters and integrated navigation. Licenciate thesis, Linköping university. Thesis No. 945.

Pitt, M. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599.

Singer, R. (1970). Estimating optimal tracking filter performance for manned maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*.

Smith, A. and Gelfand, A. (1992). Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician*, 46(2):84–88.

D

Particle Filters for System Identification of State-space Models Linear in Either Parameters or States

Authors: Thomas Schön and Fredrik Gustafsson.

Edited version of paper originally published in *proceedings of the 13th IFAC Symposium on System Identification*, pages 1287–1292, Rotterdam, The Netherlands, 2003.

Preliminary version published as Technical Report LiTH-ISY-R-2518, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Particle Filters for System Identification of State-space Models Linear in Either Parameters or States

Thomas Schön and Fredrik Gustafsson.

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

Abstract

The potential use of the marginalized particle filter for nonlinear system identification is investigated. The particle filter itself offers a general tool for estimating unknown parameters in nonlinear models of moderate complexity, and the basic trick is to model the parameters as a random walk (so called roughening noise) with decaying variance. We derive algorithms for systems which are linear in either the parameters or the states, but generally not in both. In these cases, marginalization applies to the linear part, which firstly significantly widens the scope of the particle filter to more complex systems, and secondly decreases the variance in the linear parameters/states for fixed filter complexity. This second property is illustrated on an example of a chaotic model. The particular case of freely parameterized linear state-space models, common in subspace identification approaches, is bilinear in states and parameters, and thus both cases above are satisfied. One can then choose which one to marginalize.

Keywords: System identification, Nonlinear estimation, Recursive estimation, Particle filters, Kalman filters, Bayesian estimation, marginalization, Rao-Blackwellization.

1 Introduction

In this contribution, the particle filter (Doucet *et al.*, 2001a; Gordon *et al.*, 1993) is applied to some classical system identification problems (Ljung, 1999) based on

time-varying parametric state-space models

$$z_{t+1} = f_t(z_t, \theta) + w_t^z, \tag{1a}$$
$$y_t = h_t(z_t, \theta) + e_t, \tag{1b}$$

where $z \in \mathbb{R}^{n_z}$ is the state variable, $\theta_t \in \mathbb{R}^{n_\theta}$ is the parameter vector, and $y \in \mathbb{R}^{n_y}$ is the output variable. The additive noise terms are assumed to be independent and identically distributed (i.i.d.).

First, we briefly review the problem formulation given in the accompanying paper by Gustafsson and Hriljac (2003). By augmenting the state vector with the parameters, $x_t = [z_t^T, \theta_t^T]^T$, and assuming a random walk parameter variation (of which constant parameters is a special case), we get

$$\begin{bmatrix} z_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} f_t(z_t, \theta_t) \\ \theta_t \end{bmatrix} + \begin{bmatrix} w_t^z + v_t^z \\ w_t^\theta + v_t^\theta \end{bmatrix} \tag{2a}$$
$$y_t = h_t(z_t, \theta_t) + e_t, \tag{2b}$$

where the noises are physical state noise w_t^z, state roughening noise v_t^z, parameter random walk for time-varying parameters w_t^θ and parameter roughening noise v_t^θ. The roughening noise is instrumental in the particle filter to get good performance, and is a second level design parameter. For system identification, $w_t^\theta = 0$ and v_t^θ has a variance decaying to zero, which yields converging parameter estimates. The particle filter recursively approximates the *a posteriori* density function $p(X_t|Y_t)$, where $X_t \triangleq \{x_i\}_{i=0}^t$, and the approximation converges to the true *a posteriori* density when the number of particles tends to infinity. The only problem is that the practical limit for "infinity" depends on the dimension of x_t, that is, the sum of number of parameters, θ_t, and states, z_t.

Now, if there is linear substructure available in the model this can be exploited using *marginalization*. Conceptually, marginalization means that the linear states are marginalized out. We can then apply optimal filters for the linear states and the particle filter is only applied to the truly nonlinear states. In this way, the samples in the particle filter will live in a lower dimensional space. Hence, we will intuitively obtain *more accurate estimates* for a given number of samples, since we use the optimal filters for a part of the state vector. Alternatively, we can apply the particle filter on *more complex models*. These are the practical implications of our contribution.

We will in this contribution consider the two following special cases of (1a):

1. The model is affine in the parameters and possibly nonlinear in the states,

$$f(z_t, \theta_t) = f_t^z(z_t) + A_t(z_t)\theta_t, \tag{3a}$$
$$h(z_t, \theta_t) = h_t(z_t) + C_t(z_t)\theta_t. \tag{3b}$$

2. The model is affine in the states and possibly nonlinear in the parameters,

$$f(z_t, \theta_t) = f_t^\theta(\theta_t) + A_t(\theta)z_t, \tag{4a}$$
$$h(z_t, \theta_t) = h_t(\theta_t) + C_t(\theta)z_t. \tag{4b}$$

In the subsequent two sections we will introduce the particle filter and the marginalization technique used for variance reduction. In Section 4 the models we consider are introduced and we discuss the connections to subspace identification. Section 5 is devoted to applying the marginalized particle filter to nonlinear system identification problem posed by a chaotic system. Finally, the conclusions are given in Section 6.

2 The Particle Filter

We here briefly present the theory and main algorithm. For a more intuitive presentation, see the accompanying paper (Gustafsson and Hriljac, 2003).

2.1 Recursive Bayesian Estimation

Consider systems that are described by the generic state-space model (2). The optimal Bayesian filter in this case is given below. For further details, consult (Doucet *et al.*, 2001a; Jazwinski, 1970).

Denote the observations at time t by $Y_t \triangleq \{y_i\}_{i=0}^t$. The Bayesian solution to compute the *a posteriori* density, $p(x_t|Y_t)$, of the state vector, given past observations, is given by Jazwinski (1970)

$$p(x_{t+1}|Y_t) = \int p(x_{t+1}|x_t)p(x_t|Y_t)\,dx_t, \tag{5a}$$

$$p(x_t|Y_t) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})}. \tag{5b}$$

For expressions on $p(x_{t+1}|x_t)$ and $p(y_t|x_t)$ in (5) we use the known probability densities $p_{e_t}(x)$ and $p_{v_t+w_t}(x)$, with all noises assumed independent,

$$p(x_{t+1}|x_t) = p_{v_t+w_t}\big(x_{t+1} - f(x_t)\big), \tag{6a}$$

$$p(y_t|x_t) = p_{e_t}\big(y_t - h(x_t)\big). \tag{6b}$$

2.2 Implementation

A numerical approximation to (5) is given by

$$p(x_t|Y_t) \approx \sum_{i=1}^{N} \tilde{q}_t^{(i)} \delta(x_t - x_t^{(i)}), \tag{7}$$

where $\delta(\cdot)$ is Dirac's delta function. The particles $x_t^{(i)}$ and the corresponding weights $\tilde{q}_t^{(i)}$ represent a sampled version of the *a posteriori* density $p(x_t|Y_t)$ (Doucet *et al.*, 2001a), and intuitively, the more samples the better approximation.

2.3 The Algorithm

The discussion in the previous section is summarized in the algorithm below. This is the algorithm presented in Gordon *et al.* (1993) under the name, *Bayesian bootstrap filter*.

Algorithm D.1 (The Particle Filter)

1. *Initialization: For* $i = 1, \ldots, N$, *initialize the particles,* $x_{0|-1}^{(i)} \sim p_{x_0}(x_0)$.
2. *For* $i = 1, \ldots, N$, *evaluate the importance weights* $q_t^{(i)} = p(y_t|x_{t|t-1}^{(i)})$ *and normalize* $\tilde{q}_t^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^{N} q_t^{(j)}}$.
3. *Measurement update: Resample with replacement* N *particles according to*
$$\Pr(x_{t|t}^{(i)} = x_{t|t-1}^{(j)}) = \tilde{q}_t^{(j)} \tag{8}$$
4. *Time update: For* $i = 1, \ldots, N$, *predict new particles according to*
$$x_{t+1|t}^{(i)} \sim p(x_{t+1|t}|x_{t|t}^{(i)}) \tag{9}$$
5. *Set* $t := t + 1$ *and iterate from step 2.*

The particle filter can be interpreted as a simulation-based method, i.e., N possible state trajectories, $\{x_t^{(i)}\}_{i=1}^{N}$, are simulated. Based on the measurements each trajectory is assigned a weight, $\tilde{q}_t^{(i)}$, representing the probability of that trajectory being the correct one.

3 Marginalization for Variance Reduction

Consider the case where the model is linear in some of the states. Then the Kalman filter can be used to estimate the linear states, denoted x_t^l, and the particle filter can be used to estimate the nonlinear states, denoted x_t^n. To separate the problem of estimating $p(x_t^l, x_t^n|Y_t)$ into one linear and one nonlinear problem, Bayes' theorem is used

$$p(x_t^l, X_t^n|Y_t) = p(x_t^l|X_t^n, Y_t)p(X_t^n|Y_t). \tag{10}$$

Here the density $p(x_t^l|X_t^n, Y_t)$ is given by the Kalman filter and the particle filter is used to estimate $p(X_t^n|Y_t)$. This means that the particles live in a lower-dimensional space, and it can indeed be proven (Doucet *et al.*, 2001b; Nordlund, 2002) that the variance of any function of the state and parameter is decreased or remains constant when using marginalization for a given number of particles. This technique of marginalizing out the linear state is also referred to as Rao-Blackwellization (Doucet *et al.*, 2001b).

Let the entity we want to estimate for some inference function, $g(\cdot)$, be given by

$$I(g(x_t)) = E_{p(x_t|Y_t)}[g(x_t)] = \int g(x_t)p(x_t|Y_t)dx_t. \tag{11}$$

Furthermore, let the estimate of (11) using N particles and the standard particle filter be denoted by $\hat{I}^s_N(g(x_t))$. When the marginalized particle filter is used the same estimate is denoted by $\hat{I}^m_N(g(x_t))$. Then there is a central limit theorem stating that for large N we have

$$\hat{I}^s_N(g(x_t)) \approx \mathcal{N}(I(g(x_t), R_s(N)), \tag{12a}$$

$$\hat{I}^m_N(g(x_t)) \approx \mathcal{N}(I(g(x_t), R_m(N)), \tag{12b}$$

where

$$R_s(N) \geq R_m(N). \tag{13}$$

For details concerning this result see e.g., Doucet *et al.* (1999, 2001b), or Nordlund (2002).

Asymptotically as the number of particles tend to infinity there is nothing to gain in using marginalization, since then the particle filter will provide a perfect description of $p(x^l_t, x^n_t|Y_t)$. However, since we only can use a finite number of particles it is certainly useful to marginalize and use the optimal filter, i.e., the Kalman filter, for the linear states. For details concerning the marginalized particle filter, the reader is referred to e.g., Chen and Liu (2000); Doucet *et al.* (2001b), or Nordlund (2002).

4 Models

In this section it will be shown how the particle filter can be used to estimate the nonlinear states and the Kalman filter to estimate the linear states, using the marginalization technique discussed above. All noise terms associated with the linear states are here assumed to be Gaussian, which means that the optimal estimator for the linear states/parameters is given by the Kalman filter. For the details concerning the Kalman filter equations, the state transition densities, and the likelihood functions in Algorithms D.2 and D.3 the reader is referred to Nordlund (2002). First there will be a discussion on models that are linear in the states and nonlinear in the parameters. This is followed by the reversed case, i.e., linear in the parameters and nonlinear in the states.

4.1 State-space Models Linear in the States

A state-space model linear in the states and possibly nonlinear in the parameters is written as

$$z_{t+1} = f_t^z(\theta_t) + A_t(\theta_t)z_t + w_t^z, \tag{14a}$$
$$\theta_{t+1} = \theta_t + v_t^\theta, \tag{14b}$$
$$y_t = h_t(\theta_t) + C_t(\theta_t)z_t + e_t, \tag{14c}$$

where $v_t^\theta \in \mathcal{N}(0, Q_t^{v,\theta})$ and $w_t^z \in \mathcal{N}(0, Q_t^{w,z})$[1]. Note that we can let the roughening noise v_t^z be zero when using marginalization. The *a posteriori* density will here be separated using Bayes' theorem according to

$$p(z_t, \Theta_t|Y_t) = p(z_t|\Theta_t, Y_t)p(\Theta_t|Y_t). \tag{15}$$

Note that we here consider the *a posteriori* of the complete parameter trajectory Θ_t, but only the last state vector z_t. The first density on the right hand side in (15) is given by the Kalman filter, while the second one is approximated by the particle filter. That is, we randomize particles in the parameter space according to our prior, and then each particle trajectory will be associated with one Kalman filter. The exact algorithm is given below.

Algorithm D.2 (The SIR particle filter for linear states)

1. *Initialization: For* $i = 1, \ldots, N$, *initialize the particles,* $\theta_{0|-1}^{(i)} \sim p_{\theta_0}(\theta_0)$ *and set* $\{z_{0|-1}^{(i)}, P_{0|-1}^{(i)}\} = \{\bar{z}_0, \bar{P}_0\}$.

2. *Let* $C_t^{(i)} = C_t(\theta_{t|t-1}^{(i)})$ *and* $h_t^{(i)} = h(\theta_{t|t-1}^{(i)})$. *For* $i = 1, \ldots, N$, *evaluate the importance weights*

$$q_t^{(i)} = p(y_t|\Theta_t^{(i)}, Y_{t-1}) = \mathcal{N}(h_t + C_t^{(i)} z_{t|t-1}^{(i)}, C_t^{(i)} P_{t|t-1}^{(i)} (C_t^{(i)})^T + R_t),$$

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

3. *Particle filter measurement update: Resample with replacement* N *particles according to,* $Pr(\theta_{t|t}^{(i)} = \theta_{t|t-1}^{(j)}) = \tilde{q}_t^{(j)}$.

4. *Particle filter time update and Kalman filter*

[1]The noise on the nonlinear part, here v_t^θ, can in fact have an arbitrary distribution. Similarly, The PDF $p_{\theta_0}(\theta_0)$ does not have any restrictions, since it is only used in the particle filter, the same goes for $p_{e_t}(e_t)$ if $C = 0$ in (14c). However, we leave these generalizations as a remark and assume Gaussian distributions.

(a) *Kalman filter measurement update: Let* $h_t^{(i)} = h_t(\theta_{t|t}^{(i)}), C_t^{(i)} = C_t(\theta_{t|t}^{(i)})$.

$$z_{t|t}^{(i)} = z_{t|t-1}^{(i)} + K_t^{(i)}(y_t - h_t^{(i)} - C_t^{(i)} z_{t|t-1}^{(i)}), \tag{16a}$$

$$P_{t|t}^{(i)} = P_{t|t-1}^{(i)} - K_t^{(i)} M_t^{(i)} (K_t^{(i)})^T, \tag{16b}$$

$$M_t^{(i)} = C_t^{(i)} P_{t|t-1}^{(i)} (C_t^{(i)})^T + R_t, \tag{16c}$$

$$K_t^{(i)} = P_{t|t-1}^{(i)} (C_t^{(i)})^T (M_t^{(i)})^{-1}. \tag{16d}$$

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

$$\theta_{t+1|t}^{(i)} \sim p(\theta_{t+1|t}|\Theta_t^{(i)}, Y_t) = \mathcal{N}(\theta_{t|t}^{(i)}, Q_t^{v,\theta}).$$

(c) *Kalman filter time update: Let* $A_t^{(i)} = A_t(\theta_{t+1|t}^{(i)})$ *and* $f_t^{z,(i)} = f_t^z(\theta_{t+1|t}^{(i)})$.

$$z_{t+1|t} = A_t^{(i)} z_{t|t}^{(i)} + f_t^{z,(i)}, \tag{17a}$$

$$P_{t+1|t}^{(i)} = A_t^{(i)} P_{t|t}^{(i)} (A_t^{(i)})^T + Q_t^{w,z}. \tag{17b}$$

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

Comparing the algorithms D.1 and D.2 we see that the differences are in the prediction step, which now consists of a Kalman filter update stage (split into step 4*a* and 4*c*) besides the prediction of the nonlinear states.

In some cases the same Riccati recursion can be used for all the particles, and hence a lot of computations can be saved. This occurs when the matrices A_t and C_t in (14) are independent of θ_t. In this case $P_{t|t}^{(i)} = P_{t|t}$ for all $i = 1, \ldots, N$ and hence the covariance only has to be updated once for each t. More on this can be found in Gustafsson *et al.* (2002).

4.2 State-space Models Linear in the Parameters

A state-space model that is linear in the parameters can be written as

$$z_{t+1} = f_t^z(z_t) + A_t(z_t)\theta_t + w_t^z, \tag{18a}$$

$$\theta_{t+1} = \theta_t + v_t^\theta, \tag{18b}$$

$$y_t = h_t^z(z_t) + C_t(z_t)\theta_t + e_t. \tag{18c}$$

In this case the *a posteriori* density will be split the other way around, compared to the previous section, i.e.,

$$p(Z_t, \theta_t|Y_t) = p(\theta_t|Z_t, Y_t)p(Z_t|Y_t). \tag{19}$$

The last density is approximated by the particle filter, while the first one can be solved by a Kalman filter for a parameter estimation problem in a linear regression framework. The corresponding algorithm will thus be

Algorithm D.3 (The SIR particle filter for linear parameters)

1. *Initialization: For* $i = 1, \ldots, N$, *initialize the particles,* $z_{0|-1}^{(i)} \sim p_{z_0}(z_0)$ *and set* $\{\theta_{0|-1}^{(i)}, P_{0|-1}^{(i)}\} = \{\bar{\theta}_0, \bar{P}_0\}$.
2. *Let* $h_t^{(i)} = h_t(z_{t|t-1}^{(i)}), C_t^{(i)} = C_t(z_{t|t-1}^{(i)})$. *For* $i = 1, \ldots, N$, *evaluate the importance weights*

$$q_t^{(i)} = p(y_t|Z_t^{(i)}, Y_{t-1}) = \mathcal{N}(h_t^{(i)} + C_t^{(i)}\theta_{t|t-1}, C_t^{(i)} P_{t|t-1}^{(i)} (C_t^{(i)})^T + R_t),$$

   and normalize $\tilde{q}^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^N q_t^{(j)}}$.
3. *Particle filter measurement update: Resample with replacement* N *particles according to,* $Pr(z_{t|t}^{(i)} = z_{t|t-1}^{(j)}) = \tilde{q}_t^{(j)}$.
4. *Particle filter time update and Kalman filter*
   (a) *Kalman filter measurement update: Let* $h_t^{(i)} = h_t(z_{t|t}^{(i)}), C_t^{(i)} = C_t(z_{t|t}^{(i)})$.

$$\theta_{t|t}^{(i)} = \theta_{t|t-1}^{(i)} + K_t^{(i)}(y_t - h_t^{(i)} - C_t^{(i)}\theta_{t|t-1}^{(i)}), \tag{20a}$$
$$P_{t|t}^{(i)} = P_{t|t-1}^{(i)} - K_t^{(i)} M_t^{(i)} (K_t^{(i)})^T, \tag{20b}$$
$$M_t^{(i)} = C_t^{(i)} P_{t|t-1}^{(i)} (C_t^{(i)})^T + R_t, \tag{20c}$$
$$K_t^{(i)} = P_{t|t-1}^{(i)} (C_t^{(i)})^T (M_t^{(i)})^{-1}. \tag{20d}$$

   (b) *Particle filter time update: Let* $f_t^{z,(i)} = f_t^z(z_{t+1|t}^{(i)})$ *and* $A_t^{(i)} = A_t(z_{t+1|t}^{(i)})$. *For* $i = 1, \ldots, N$, *predict new particles,*
   $z_{t+1|t}^{(i)} \sim p(z_{t+1|t}|Z_t^{(i)}, Y_t) = \mathcal{N}(f_t^{z,(i)} + A_t^{(i)}\theta_{t|t}^{(i)}, A_t^{(i)} P_{t|t}^{(i)} (A_t^{(i)})^T + Q_t^{w,z})$.
   (c) *Kalman filter time update: Let* $f_t^{z,(i)} = f_t^z(z_{t+1|t}^{(i)})$ *and* $A_t^{(i)} = A_t(z_{t+1|t}^{(i)})$.

$$\theta_{t+1|t}^{(i)} = \theta_{t|t}^{(i)} + L_t^{(i)}(z_{t+1|t}^{(i)} - f_t^{z,(i)} - A_t^{(i)}\theta_{t|t}^{(i)}), \tag{21a}$$
$$P_{t+1|t}^{(i)} = P_{t|t}^{(i)} + Q_t^{v,\theta} - L_t^{(i)} N_t^{(i)} (L_t^{(i)})^T, \tag{21b}$$
$$N_t^{(i)} = A_t^{(i)} P_{t|t}^{(i)} (A_t^{(i)})^T + Q_t^{w,z}, \tag{21c}$$
$$L_t^{(i)} = P_{t|t}^{(i)} (A_t^{(i)})^T (N_t^{(i)})^{-1}. \tag{21d}$$

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

The measurements used in the Kalman filter are thus the "normal" measurements, y_t, and the predicted state trajectory, $z_{t+1|t}$, i.e., the samples from the particle filter. Step 4*c* in the current algorithm contains a measurement update, using the

prediction (since this contains information about θ_t) from the particle filter, and a time update.

An interesting special case of the two different model types discussed above is when we consider "the intersection" of the two types, i.e., a model that is bilinear in the states, z_t, and in the parameters, θ_t.

A particular case of interest is a general state-space model in innovation form

$$z_{t+1} = A(\theta_t)z_t + K(\theta_t)e_t \tag{22a}$$
$$y_t = C(\theta_t)z_t + e_t, \tag{22b}$$

where the parameters enter linearly in A, K, and C. The *a posteriori* density will here be split according to (19). One popular approach here is so called subspace identification (van Overschee and Moor, 1996). This class of algorithms usually perform very well and provides consistent estimates. One limitation is that it is hard to give the *a posteriori* density of the parameters, even in the Gaussian case, and this is perhaps where the particle filter can help. This case is mentioned to show the relation to classical system identification problems.

Assume, to avoid ambiguities in the state coordinates, an observer canonical form and scalar output, where $C = [1, 0, \dots 0]$ and that all parameters in A and K are unknown. Then, given the state trajectory and measurement, we have from (22) the linear regression $z_{t+1} = Az_t + K(y_t - [1, 0, \dots 0]z_t)$. This regression problem has to be solved for each particle $z_t^{(i)}, i = 1, \dots, N$.

In the case where there are more states to be estimated than parameters, i.e., $\dim z_t > \dim \theta$ it is better to split the density $p(Z_t, \theta_t|Y_t)$ in (19) the other way around, i.e., as in (15). This time, a Kalman filter estimating the states, z_t, for each particle, $\theta_t^{(i)}$, is needed. In this way the dimension of the state estimated by the particle filter is kept as low as possible. An example where this situation typically occurs is in gray-box identification Ljung (1999).

5 Chaos Example

The ideas presented in this article will be illustrated using the following chaotic model

$$z_{t+1} = (1 - z_t)z_t\theta + w_t, \tag{23a}$$
$$y_t = z_t + e_t, \tag{23b}$$

where z_t is the state variable, y_t is the measurement, θ is the unknown parameter, w_t is the process noise, and e_t is the measurement noise. Both these noise densities are Gaussian distributed. The aim is to recursively estimate both the state, z_t, and the parameter, θ. This model is linear in the time-invariant parameter θ and nonlinear in the state z_t. This fits our framework, according to Section 4.2 and hence Algorithm D.3 can be applied. This problem has also been studied in Gustafsson and Hriljac (2003), where the particle filter was directly applied to the

augmented state $x_t = [z_t,\, \theta_t]^T$. Model (23) can be written on the form (18), i.e.,

$$z_{t+1} = A_t(z_t)\theta_t + w_t^z + v_t^z, \tag{24a}$$

$$\theta_{t+1} = \theta_t + v_t^\theta, \tag{24b}$$

$$y_t = h_t(z_t) + e_t, \tag{24c}$$

where $A_t(z_t) = (1 - z_t)z_t$ and $h_t(z_t) = z_t$. The two noises $v_t^z \sim \mathcal{N}(0,\, Q_t^{v,z})$ and $v_t^\theta \sim \mathcal{N}(0,\, Q_t^{v,\theta})$ are roughening noises. Furthermore, $e_t \sim \mathcal{N}(0, R_t)$.

In the simulations, two different particle filters were used, the standard particle filter, Algorithm D.1, applied to the augmented state vector, x_t, and the marginalized particle filter according to Algorithm D.3. The true value of θ is 3.92, and the initial guess is $\theta_{0|-1} \sim \mathcal{N}(3.83, 0.04)$. The initial state is $z_0 \sim \mathcal{N}(0, 1)$. We do not use any process noise, however we have roughening noises $Q_0^{v,z} = Q_0^{v,\theta} = 10^{-2}$, which is decreased at each time step, according to Gustafsson and Hriljac (2003). The measurement noise has variance $R_t = 10^{-5}$, and we have used 200 Monte Carlo simulations. In Figure 1 the filtered estimates of θ are shown using these two algorithms for 150, 1000, and 10000 particles respectively. In order to make

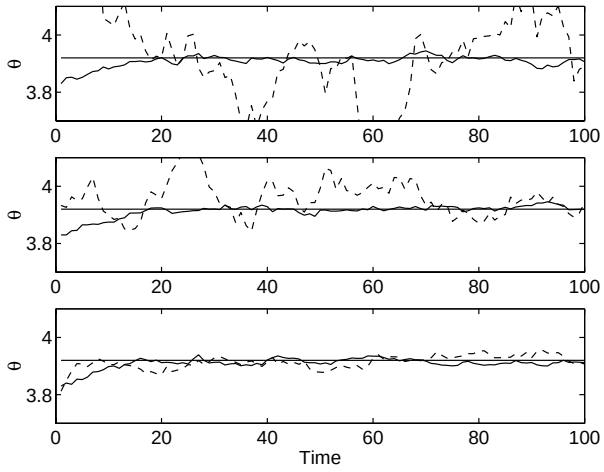


Figure 1 *Estimates of θ using the standard (dashed) and the marginalized (solid) particle filters. The true θ is shown using a solid line. Top plot* - 150 *particles, middle* - 1000 *particles, bottom* - 10000 *particles.*

the difference more apparent the Root Mean Square Error (RMSE) is plotted in Figure 2 as a function of the number of particles used in the simulations. Note that the RMSE values are calculated from time 50. In that way the transient effects are not included in the RMSE values. According to (13) the estimates should be

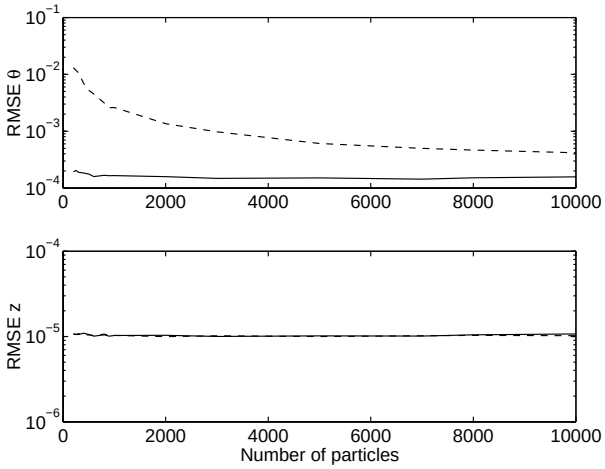


Figure 2 *RMSE values for $\hat{\theta}$ (top) and $\hat{z}_t$ (bottom) as a function of the number of particles used. Notice that a log-scale has been used in the plots, and that a dashed line has been used for the standard particle filter and a solid line for the marginalized particle filter.*

better or the same when we use the marginalized particle filter. From Figure 2 we can see that this is indeed the case. It is only the estimate of the linear part, θ, that is improved, this is also consistent with the theory, see e.g., Nordlund (2002) for the theoretical details. That this is true in the simulations is apparent by Figure 2, from which it is clear that the estimate of the linear part (top) clearly is better using the marginalized particle filter. The estimate of the nonlinear part, z_t, has the same quality. Of course if we could use an infinite number of particles the results using the standard and the marginalized particle filter would have been the same, since then the particle filter would be able to provide an arbitrarily good estimate of $p(x_t|Y_t)$. We can see indications of this fact in the top plot in Figure 2, since the more particles that are used the closer the estimates get to each other.

6 Conclusions

The potential use of particle filtering for system idenfication of unknown parameters in nonlinear systems was explained in the accompanying paper Gustafsson and Hriljac (2003). Here, we have proposed the use of marginalized particle filters. More specifically, we studied the cases where the model is either linear in the states and nonlinear in the parameters, or nonlinear in the states and linear in the parameters. The algorithms were given for these two cases. It is straightforward to give the algorithm for an arbitrary mix of linear and nonlinear states and parameters. The

advantage of marginalization is that one can apply the filter to larger problems with more states and parameters, or that fewer particles and thus less filter complexity is needed for a given performance. Finally an example was given, which illustrates the improvement in estimation performance compared to using a standard particle filter.

References

Chen, R. and Liu, J. (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society*, 62(3):493–508.

Doucet, A., de Freitas, N., and Gordon, N., editors (2001a). *Sequential Monte Carlo Methods in Practice*. Springer Verlag.

Doucet, A., Gordon, N., and Krishnamurthy, V. (1999). Particle filters for state estimation of jump Markov linear systems. Technical Report CUED/F-INFENG/TR 359, Signal Processing Group, Department of Engineering, University of Cambridge, Trupington street, CB2 1PZ Cambridge.

Doucet, A., Gordon, N., and Krishnamurthy, V. (2001b). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624.

Gordon, N., Salmond, D., and Smith, A. (1993). A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113.

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J. (2002). Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437.

Gustafsson, F. and Hriljac, P. (2003). Particle filters for prediction of chaos. In *proceedings of the 13th IFAC Symposium on System Identification*, pages 1281–1286, Rotterdam, The Netherlands.

Jazwinski, A. (1970). *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York.

Ljung, L. (1999). *System identification, Theory for the user*. System sciences series. Prentice Hall, Upper Saddle River, NJ, 2nd edition.

Nordlund, P.-J. (2002). Sequential Monte Carlo filters and integrated navigation. Licenciate thesis, Linköping university. Thesis No. 945.

van Overschee, P. and Moor, B. D. (1996). *Subspace identification for linear systems - theory, implementation, applications*. Kluwer Academic Publishers.

A

Notation

Note that all vectors are column vectors. In general lower case letters are used to denote vector valued and scalar variables, and upper case letters are used for matrix valued variables. However, there might be exceptions from these general rules due to conventions.

Operators

$\arg\max_x f(x)$	value of x the maximizes $f(x)$
$\min$	minimize
$\max$	maximize
$\mathrm{Cov}[x]$	covariance of the random variable x
$\mathrm{Var}[x]$	variance of the random variable x
$\mathrm{E}[x]$	expected value of the random variable x
$\mathrm{E}[\cdot\|y]$	conditional expectation given y
$\det[A]$	determinant of the matrix A
A^T	transpose of the matrix A
A^{-1}	inverse of the matrix A
$r_i[P]$	the i^{th} row degree of a polynomial matrix $P(s)$

$\text{diag}(a)$	a diagonal matrix with a as diagonal entry
$\|x\|_A^2$	weighted vector norm, $\|x\|_A^2 = x^T Ax$
$\|\cdot\|$	absolute value
$\dot{x}(t)$	time derivative of x
$\mathcal{R}(B)$	range of the matrix B
$r_i[P]$	i^{th} row degree of the polynomial matrix $P(s)$
$\deg[f(x)]$	degree of the function $f(x)$
$\triangleq$	equal by definition
$\propto$	proportional to
$\in$	belongs to
$\forall$	for all
$\delta(\cdot)$	Dirac's delta function
δ_{kl}	Kronecker's delta function
$\Pr(x = a)$	probability that the random variable x equals a
$p_x(x)$	probability density function of x
$p(x) \triangleq p_x(x)$	short form of above
$p_{x,y}(x, y)$	joint probability density function of x and y
$p(x, y) \triangleq p_{x,y}(x, y)$	short form of above
$p_{x\|y}(x\|y)$	conditional probability density function of x given y
$p(x\|y) \triangleq p_{x\|y}(x\|y)$	short form of above
$\mathrm{E}\,\cdot\|b$	conditional expectation given b

Symbols

L	length of the sliding window
N	number of particles
N_{eff}	effective sample size
t	current time
T_s	sample time
$\bar{x}_0$	initial value for the state x
$\bar{P}_0$	initial value for the covariance P
θ	parameter vector, dimension $= n_\theta$
e_t	measurement noise, dimension $= n_e$
$q(X_t)$	importance weights
$\tilde{q}_t^{(i)}$	normalized importance weights
w_t	state noise, dimension $= n_w$
X_t	stacked vector of the process x_t from the initial time, until time t
$X_{k:t}$	stacked vector of the process x_t from time k, until time t
$\hat{x}$	estimator or estimate of x, determined by the context
$\hat{x}_{t\|t}$	estimate of x at time t, given the information available at time t, i.e., the filtered estimate

$\hat{x}_{t+k\|t}$	estimate of x at time $t+k$, given the information available at time t. $k<0$ means smoothed estimate, and $k>0$ means predicted estimate
x_t	state vector at time t, dimension $= n_x$
x_t^l	linear state variable at time t, dimension $= n_{x^l}$
x_t^n	nonlinear state variable at time t, dimension $= n_{x^n}$
y_t	measurement vector at time t, dimension $= n_y$
K_t	Kalman gain at time t
P_t	covariance matrix at time t
Q_t	covariance matrix for the process noise w_t
R_t	covariance matrix for the measurement noise e_t
φ_t	regression vector at time t
$\mathbb{R}^n$	the set of real numbers in n dimensions
$\mathbb{R}^+$	the set of positive real numbers
$\mathbb{N}$	the set of natural numbers, i.e., $\{0,1,2,\dots\}$
$l(x)$	likelihood function
$f(\cdot)$	equations for the system model
$h(\cdot)$	equations for the measurement model
$g(\cdot)$	inference function
$\hat{p}_N(\cdot)$	probability density function approximated using N samples
$\pi(\cdot)$	importance density
$K(\cdot)$	kernel function
$G(e^{i\omega})$	transfer function
$\mathcal{N}(m,Q)$	normal distribution with mean m and covariance Q

Abbreviations and acronyms

a.s.	almost sure
i.i.d.	independent and identically distributed
pdf	probability density function
s.t.	subject to
w.r.t.	with respect to
ARX	AutoRegressive with eXternal input
DAE	Differential Algebraic Equation
DGPS	Differential GPS
EKF	Extended Kalman Filter
GPS	Global Positioning System
HMM	Hidden Markov Model
INS	Inertial Navigation System
LMI	Linear Matrix Inequality
LS	Least Squares
MAP	Maximum A Posteriori

MCMC	Markov Chain Monte Carlo
MFD	Matrix Fraction Description
MHE	Moving Horizon Estimation
ML	Maximum Likelihood
MPC	Model Predictive Control
MV	Minimum Variance
ODE	Ordinary Differential Equation
QP	Quadratic Program
RLS	Recursive Least Squares
RMSE	Root Mean Square Error
SIS	Sequential Importance Sampling
SIR	Sequential Importance Resampling
SOCP	Second-Order Cone Program
SVD	Singular Value Decomposition
TAP	Terrain Aided Positioning
WLS	Windowed Least Squares
YALMIP	Yet Another LMI Parser

Index

Tekn. lic. Dissertations
Division of Automatic Control and Communication Systems
Linköping University

P. Andersson: Adaptive Forgetting through Multiple Models and Adaptive Control of Car Dynamics. Thesis No 15, 1983.

B. Wahlberg: On Model Simplification in System Identification. Thesis No 47, 1985.

A. Isaksson: Identification of Time Varying Systems and Applications of System Identification to Signal Processing. Thesis No 75, 1986.

G. Malmberg: A Study of Adaptive Control Missiles. Thesis No 76, 1986.

S. Gunnarsson: On the Mean Square Error of Transfer Function Estimates with Applications to Control. Thesis No 90, 1986.

M. Viberg: On the Adaptive Array Problem. Thesis No 117, 1987.

K. Ståhl: On the Frequency Domain Analysis of Nonlinear Systems. Thesis No 137, 1988.

A. Skeppstedt: Construction of Composite Models from Large Data-Sets. Thesis No 149, 1988.

P. A. J. Nagy: MaMiS: A Programming Environment for Numeric/Symbolic Data Processing. Thesis No 153, 1988.

K. Forsman: Applications of Constructive Algebra to Control Problems. Thesis No 231, 1990.

I. Klein: Planning for a Class of Sequential Control Problems. Thesis No 234, 1990.

F. Gustafsson: Optimal Segmentation of Linear Regression Parameters. Thesis No 246, 1990.

H. Hjalmarsson: On Estimation of Model Quality in System Identification. Thesis No 251, 1990.

S. Andersson: Sensor Array Processing; Application to Mobile Communication Systems and Dimension Reduction. Thesis No 255, 1990.

K. Wang Chen: Observability and Invertibility of Nonlinear Systems: A Differential Algebraic Approach. Thesis No 282, 1991.

J. Sjöberg: Regularization Issues in Neural Network Models of Dynamical Systems. Thesis No 366, 1993.

P. Pucar: Segmentation of Laser Range Radar Images Using Hidden Markov Field Models. Thesis No 403, 1993.

H. Fortell: Volterra and Algebraic Approaches to the Zero Dynamics. Thesis No 438, 1994.

T. McKelvey: On State-Space Models in System Identification. Thesis No 447, 1994.

T. Andersson: Concepts and Algorithms for Non-Linear System Identifiability. Thesis No 448, 1994.

P. Lindskog: Algorithms and Tools for System Identification Using Prior Knowledge. Thesis No 456, 1994.

J. Plantin: Algebraic Methods for Verification and Control of Discrete Event Dynamic Systems. Thesis No 501, 1995.

J. Gunnarsson: On Modeling of Discrete Event Dynamic Systems, Using Symbolic Algebraic Methods. Thesis No 502, 1995.

A. Ericsson: Fast Power Control to Counteract Rayleigh Fading in Cellular Radio Systems. Thesis No 527, 1995.

M. Jirstrand: Algebraic Methods for Modeling and Design in Control. Thesis No 540, 1996.

K. Edström: Simulation of Mode Switching Systems Using Switched Bond Graphs. Thesis No 586, 1996.

J. Palmqvist: On Integrity Monitoring of Integrated Navigation Systems. Thesis No 600, 1997.

A. Stenman: Just-in-Time Models with Applications to Dynamical Systems. Thesis No 601, 1997.

M. Andersson: Experimental Design and Updating of Finite Element Models. Thesis No 611, 1997.

U. Forssell: Properties and Usage of Closed-Loop Identification Methods. Thesis No 641, 1997.

M. Larsson: On Modeling and Diagnosis of Discrete Event Dynamic systems. Thesis No 648, 1997.

N. Bergman: Bayesian Inference in Terrain Navigation. Thesis No. 649, 1997.

V. Einarsson: On Verification of Switched Systems Using Abstractions. Thesis No. 705, 1998.

J. Blom, F. Gunnarson: Power Control in Cellular Radio Systems. Thesis No. 706, 1998.

P. Spångéus: Hybrid Control using LP and LMI methods – Some Applications. Thesis No. 724, 1998.

M. Norrlöf: On Analysis and Implementation of Iterative Learning Control. Thesis No. 727, 1998.

A. Hagenblad: Aspects of the Identification of Wiener Models. Thesis no 793, 1999.

F. Tjärnström: Quality Estimation of Approximate Models. Thesis no 810, 2000.

C. Carlsson: Vehicle Size and Orientation Estimation Using Geometric Fitting. Thesis no 840, 2000.

J. Löfberg: Linear Model Predictive Control: Stability and Robustness. Thesis no 866, 2001.

O. Härkegård: Flight Control Design Using Backstepping. Thesis no 875, 2001.

J. Elbornsson: Equalization of Distortion in A/D Converters. Thesis No. 883, 2001.

J. Roll: Robust Verification and Identification of Piecewise Affine Systems. Thesis No. 899, 2001.

I. Lind: Regressor Selection in System Identification using ANOVA. Thesis No. 921, 2001.

R. Karlsson: Simulation Based Methods for Target Tracking. Thesis No. 930, 2002.

P-J. Nordlund: Sequential Monte Carlo Filters and Integrated Navigation. Thesis No. 945, 2002.

M. Östring: Identification, Diagnosis, and Control of a Flexible Robot Arm. Thesis No. 948, 2002.

C. Olsson: Active Engine Vibration Isolation using Feedback Control. Thesis No. 968, 2002.

J. Jansson: Tracking and Decision Making for Automotive Collision Avoidance. Thesis No. 965, 2002.

N. Persson: Event Based Sampling with Application to Spectral Estimation. Thesis No. 981, 2002.

D. Lindgren: Subspace Selection Techniques for Classification Problems. Thesis No. 995, 2002.

E. Geijer Lundin: Uplink Load in CDMA Cellular Systems. Thesis No. 1045, 2003.

M. Enqvist: Some Results on Linear Models of Nonlinear Systems. Thesis No. 1046, 2003.