View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Aaltodoc Publication Archive

Ville Väänänen

Gaussian filtering and smoothing based parameter estimation in nonlinear models for sequential data

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo October 29, 2012

Thesis supervisor:

Prof. Jouko Lampinen

Thesis advisor:

D.Sc. (Tech.) Simo Särkkä

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Ville Väänänen

Title: Gaussian filtering and smoothing based parameter estimation in nonlinear models for sequential data

Date: October 29, 2012 Language: English Number of pages:7+70

Department of Biomedical Engineering and Computational Science

Professorship: Computational and Cognitive Biosciences Code: S-114

Supervisor: Prof. Jouko Lampinen

Advisor: D.Sc. (Tech.) Simo Särkkä

State space modeling is a widely used statistical approach for sequential data. The resulting models can be considered to contain two interconnected estimation problems: that of the dynamic states and that of the static parameters. The difficulty of these problems depends critically on the linearity of the model, with respect to the states, the parameters or both.

In this thesis we show how to obtain maximum likelihood and maximum a posteriori estimates for the static parameters. Two methods are considered: gradient based nonlinear optimization of the marginal log-likelihood and expectation maximization. The former requires the filtering distributions and the latter both the filtering and the smoothing distributions. When closed form solutions to these distributions are unavailable, we apply efficient Gaussian filtering based methods to obtain approximations.

The resulting parameter estimation algorithms are demonstrated by a linear target-tracking model with simulated data and a nonlinear stochastic resonator model with photoplethysmograph data.

Keywords: Parameter estimation, Sequential data, Nonlinear state space models, Expectation maximization, Quasi-Newton optimization

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Ville Väänänen

Työn nimi: Gaussiseen suodatukseen ja siloitukseen perustuva parametrien estimointi epälineaarisissa aikasarjamalleissa

Päivämäärä: October 29, 2012 Kieli: Englanti Sivumäärä:7+70

Lääketieteellisen tekniikan ja laskennalisen tieteen laitos

Professuuri: Laskennallinen ja kognitiivinen biotiede Koodi: S-114

Valvoja: Prof. Jouko Lampinen

Ohjaaja: TkT Simo Särkkä

Tila-avaruusmallinnus on eräs laajalti käytetty aikasarjojen mallinnusmenetelmä. Tila-avaruusmallin voidaan ajatella sisältävän kaksi keskenään vuorovaikkuteista estimointiongelmaa: dynaamisten tilojen estimointi sekä staattisten parametrien estimointi. Näiden estimointiongelmien vaikeuteen vaikuttaa erityisen paljon mallin lineaarisuus – sekä tilojen että parametrien suhteen.

Tässä diplomityössä näytämme, kuinka staattisia parametrejä voidaan estimoida suurimman uskottavuuden estimaattorilla tai posteriorijakauman maksimoivalla estimaattorilla. Analysoimme kahta eri menetelmää: uskottavuusfunktion gradienttipohjaista epälineerista optimointia sekä expectation maximization algoritmiä. Näistä ensimmäinen vaatii suodinjakaumien ja jälkimmäinen sekä suodin- että siloitusjakaumien ratkaisemista. Mikäli näitä jakaumia ei voida ratkaista suljetussa muodossa, käytämme tehokkaita Gaussiseen suodatukseen perustuvia menetelmiä niiden likimääräiseen ratkaisemiseen.

Lopputuloksina saatuja parametriestimointimenetelmiä sovelletaan ensin lineaarisessa kohteenseurantamallissa simuloidulla datalla ja sen jälkeen epälineaarisessa stokastisessa resonaattorimallissa fotopletysmografidatalla.

Avainsanat: Parametrien estimointi, Aikasarjat, Epälineaariset tila-avaruusmallit, EM, Kvasi-Newton optimointi

Preface

This master's thesis was the culmination of the learning which eventually took place while I was working in the Bayesian Statistical Methods group in the Department of Biomedical Engineering and Computational Science at Aalto University, Finland.

I wish to express my sincere gratitude for the guidance and expert advice offered by my instructor D.Sc. Simo Särkkä. I am also deeply grateful to my supervisor Prof. Jouko Lampinen for his patience in the delicate process which was the completion of this thesis. Most certainly I am indebted to my dear colleague M.Sc. Arno Solin who generously offered both his time and considerable expertise for making this thesis better.

Furthermore I would like to thank M.Sc. Tomi Peltola, M.Sc. Mudassar Abbas and M.Sc. Jouni Hartikainen for their delightful company and inspirational coffee-room discussions. Many other dear friends were also instrumental for this thesis to materialize, not least by helping to remind that sometimes it is more efficient to take a break than to continue.

Finally, I would like to extend my sincere appreciation to my family whose love, support and understanding I feel so privileged to have.

Otaniemi, October 29, 2012

Ville Väänänen

Contents

Symbols and abbreviations

General notation

$\mathbf{Z}$	Matrix (bold uppercase letter)
$\mathbf{Z}^\mathsf{T}$	Transpose of matrix $\mathbf{Z}$
$\mathbf{I}$	Identity matrix
$\mathbf{z}$	Column vector (bold lowercase letter)
$\mathbf{z}^\mathsf{T}$	Row vector
$\mathbf{z}_{1:T}$	Set of vectors $\{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$
$\boldsymbol{\theta}$	Parameter
$p(\mathbf{x} \mid \mathbf{y})$	Conditional PDF of $\mathbf{x}$ given $\mathbf{y}$
$\mathbf{m}_{k\mid k-1}$	Conditional value of $\mathbf{m}_k$ given measurements up to step $k-1$
$\mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P})$	Gaussian PDF of $\mathbf{x}$ with mean $\mathbf{m}$ and covariance matrix $\mathbf{P}$
$\mathbb{N}, \mathbb{R}, \mathbb{C}$	The fields of natural, real and complex numbers

Abbreviations

AR	Autoregressive
BFGS	Broyden–Fletcher–Goldfarb–Shanno
CKF	Cubature Kalman filter
CKS	Cubature Kalman smoother
DAG	Directed acyclic graph
ECG	Expectation–conjugate–gradient
EKF	Extended Kalman filter
EM	Expectation maximization
fMRI	Functional magnetic resonance imaging
gEM	Generalized expectation maximization
GHKF	Gauss–Hermite Kalman filter
HMM	Hidden Markov model
MAP	Maximum a posteriori
MCMC	Markov chain Monte Carlo
MEG	Magnetoencephalography
ML	Maximum likelihood
PDF	Probability density function
PMCMC	Particle Markov chain Monte Carlo
RTS	Rauch–Tung–Striebel
SMC	Sequential Monte Carlo
SSM	State space model
VB	Variational Bayes
vEM	Variational expectation maximization

1 Introduction

Modeling temporal data is of great interest in numerous branches of science. Since the passage of time is so deeply embedded in our experience of the world, it is understandable that many scientific questions are posed in a *dynamic* setting. Arguably, the very concept of life implies variation in time and so as an example of a dynamic system we can consider any biological being, such as ourselves. Measuring heart rate or neuronal activity or any *biosignal*, with any of the available technologies, produces sequential data (see, e.g., Särkkä et al., 2012, for a recent application). A classic example of a dynamic system is *target tracking*, where based on a sequence of noisy range or angular measurements from a measurement intrument, typically a radar, we would like to continuously estimate the true position and velocity of the target (Bar-Shalom, Li, & Kirubarajan, 2004; Godsill, Vermaak, Ng, & Li, 2007).

Let us assume the existence of a sequential dataset which is a result of making *measurements* on a *system* of interest. In order to answer questions of interest about the system quantitatively, the system and the measurements should be mathematically *modeled*. The class of mathematical models for dynamical systems we will be concerned with are known as *state space models* (SSMs). Important characteristics of SSMs are *stochasticity* and decoupling of the system dynamics and the measurements. At any instant, the system is thought to be in a certain finite-dimensional *state*. The state summarizes enough information about the system so that it is possible to formulate the system state at the next instant as a function of the current state and *process noise*. However the state is *hidden* (or *latent*) and the inference on the state has to be made entirely based on the measurements. Often some components of the measurements would otherwise translate directly to corresponding components of the state, except that the measurements are always assumed to be *noisy*. As an example, in target tracking the state should contain at least the location and the velocity of the target.

The stochasticity forces us to assume a probabilistic framework. In this thesis the viewpoint is decidedly *Bayesian*. In Bayesian statistics, ideally, the complete answer is always the *posterior probability distribution*, meaning the joint probability distribution of the random variables of interest given the measurements. Thus instead of answering with a single value or a value with error bounds, the answer is the probability density function of the interesting quantity given the data. It is important to highlight, however, that Bayesian statistics can be used to treat many kinds of uncertainty (as pointed out in e.g. Särkkä, 2012). For example, the instruments

used to obtain the measurements are a source of uncertainty related to randomness, whereas our uncertainty regarding the model and its parameters implies another kind of uncertainty. Both kinds of uncertainties can be quantified with Bayesian statistics and thus applying statistical methods to a problem does not imply that the problem is actually random.

SSMs are a general framework and in any specific application prior knowledge of the system has to be brought in. This prior knowledge is not necessarily very specific, for example in ballistic target tracking it might include the assumption that Newton's laws are applicable. The mathematical form of the dependence between the measurements and the state has to be formulated as well as the dependence of the state on its predecessors. Usually one is able only to specify the *parametric* form for these equations. This results in a model with a set of unknown parameters, denoted with $\boldsymbol{\theta}$. In the Bayesian framework, $\boldsymbol{\theta}$ is a random variable with some prior probability distribution $p(\boldsymbol{\theta})$. In order to complete the model, $\boldsymbol{\theta}$ needs to be estimated based on some available training data, the same sequential dataset we assumed earlier. This is sometimes, at least in control engineering, known as *system identification*. In this thesis, it is assumed that the parameters are static, that is independent of time. This is then an important distinction between the states and the parameters, in this thesis.

In general, assuming the aforementioned distinction between parameters and states, there are two separate but interconnected estimation problems in SSMs: that of the states and that of the parameters. The interest might lie in either one or both, depending on the model. Traditionally, state estimation, given measurements up to the current instant, is known as *filtering*. The term can be thought to relate to the idea of filtering the noise out of the measurements in order to observe the states. Given a batch of measurements, state estimation is called *smoothing*. In order to engage in smoothing, the batch of measurements needs to be collected in its entirety, making smoothing an *offline* procedure. Filtering, on the other hand, is *online*, meaning the estimates can be updated every time a new measurement arrives.

A distinction is drawn in this thesis between *linear* and *nonlinear* models. The linear model can be thought of as a special case of the nonlinear model, so that linear models could be implicitly covered by only considering nonlinear models. The distinction is useful for the simple reason that in the linear case closed form solutions exist. The Bayesian solution of the filtering problem for a linear system with additive Gaussian noise is given by the celebrated *Kalman filter* (Kalman,

1960).

Our focus in this thesis is in the static parameter estimation problem for the nonlinear case. Depending on the method, this requires either the filtering or filtering and smoothing solutions of the state estimation problem. Thus the state and parameter estimation problems are inherently connected. For reasons of computational complexity, we will not try to pursue the complete Bayesian solution of finding the posterior distribution. We will settle for a point estimate called the *maximum a posteriori* (MAP), which is the mode (the value of the parameter giving the maximum) of the posterior distribution (Gelman, Carlin, Stern, & Rubin, 2004). Under an assumed *uniform* prior distribution, the MAP estimate becomes equivalent with the *maximum likelihood* (ML) estimate. The philosophical difference between these two might be fundamental, but as will be seen, from the perspective of the estimation equations it is not.

Figure 1 illustrates the concepts of states, measurements and parameters. In Figure 1a, we have simulated a simple first order autoregressive (AR(1)), or first order *random walk*, model for $T = 150$ steps. The states, x_k, are unidimensional and are denoted with a continuous line. This reflects the fact, that commonly the system of interest operates in *continuous* time. The measurements, y_k, are denoted with crosses which reflects the fact the measurements, our sequential dataset, are almost always discrete. The model in Figure 1 has a parameter θ and the simulation in Figure 1a is made with $\theta = 1$. In Figure 1b, we have tried to estimate θ, based only on the measurements. We have drawn a curve, which can be described as the likelihood function or the unnormalized posterior probability distribution with a uniform prior distribution. We can see that if we choose as our estimate the mode of this curve, denoted with a star, our estimate would be close to the true value.

We begin with the background, where SSMs are covered in necessary detail, Bayesian optimal filtering and smoothing equations are derived and the role of the static parameters is elaborated on. Following the background, in Section 3 we focus on state estimation, first for linear and then for nonlinear systems. The Kalman filter is introduced here as is the concept of Gaussian filtering, a deterministic approximation used for nonlinear systems. The fourth section is concerned with the two methods of parameter estimation we are comparing: gradient based nonlinear optimization and the Expectation Maximization (EM) algorithm. Our goal here is to present the underlying ideas, the resulting equations and the most helpful literary references for one to be able to actually implement these methods.

The theoretical analysis is sufficiently detailed in order to draw some conclusions

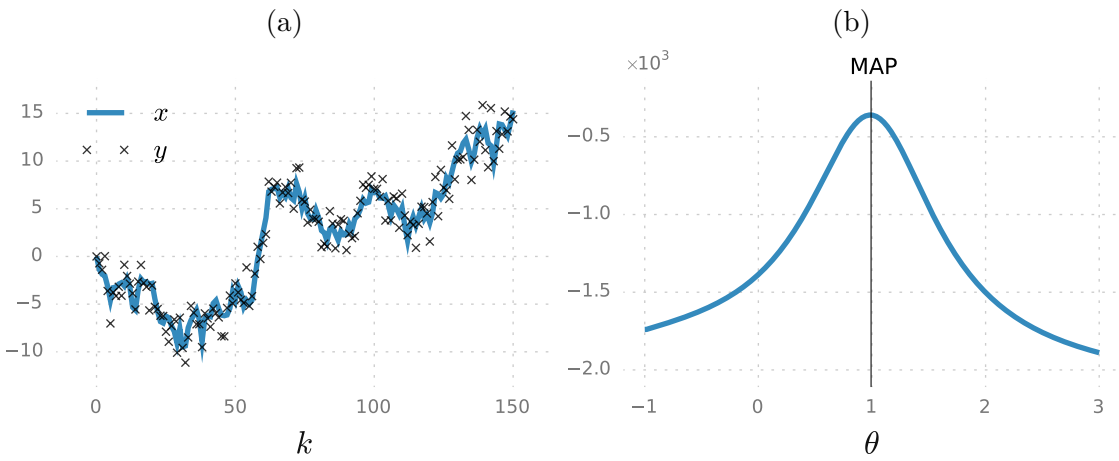

Figure 1: (a) A simulation of a first order random walk model in Section 1, with parameter $\theta = 1$.The noisy measurements are denoted with crosses. (b) A plot of the unnormalized posterior probability distribution of θ, given the simulated measurements and assuming a uniform prior.

about the behavior of the parameter estimation methods in the results section. The results section has two subsections: a target tracking application with simulated data and a biomedical signal processing application with real world data.

2 Background

2.1 State space models

State space models (SSMs) provide a unified probabilistic methodology for modeling sequential data (Ljung & Glad, 1994; Durbin & Koopman, 2012; Cappé, Moulines, & Rydén, 2005; Barber, Cemgil, & Chiappa, 2011). Sequential data arise in numerous applications, typically in the form of time-series measurements. Modern time-series data arise often in the context of medical imaging, for example in the case of functional magnetic resonanse imaging (fMRI) or magnetoencephalography (MEG). However it is not necessary for the sequence index to have a temporal meaning. In probabilistic terms, a time-series can be described by a *stochastic process* $\mathbf{y} = \{\mathbf{y}(t) : t \in \mathcal{T}\}$, where $\mathbf{y}(t)$ is a random variable and $\mathcal{T} \subseteq \mathbb{R}$ for continuous time or $\mathcal{T} \subseteq \mathbb{N}$ for discrete time sequences. In this thesis we will only be concerned with discerete time processes and we write $\mathbf{y}_{1:k} \equiv \{\mathbf{y}_1, \ldots, \mathbf{y}_k\} \equiv \{\mathbf{y}(t_1), \ldots, \mathbf{y}(t_k)\}$.

A fundamental question in probabilistic models for sequential data is how to model the dependence between variables. It is infeasible to assume that every random variable in the process depends on all the others. Thus it is common to assume a *Markov chain*, where the distribution of the process at the current timestep depends only on the probability distribution in the previous timestep. A further assumption in SSMs is that the process of interest, the dynamic process $\mathbf{x}$, is not directly observed but only through another stochastic process, the *measurement process* $\mathbf{y}$. Since $\mathbf{x}$ is not observed, SSMs belong to the class of *latent variable models*. Sometimes, as in Cappé et al. (2005), SSMs are called *hidden Markov models* (HMM) but usually this implies that the sample space of $\mathbf{x}$ is discrete. Yet another term for a quite general subclass of SSMs is *dynamic Bayesian networks* (DBNs). These and some connections to classical time-series modeling approaches are discussed in Murphy (2002).

An important characteristic of SSMs is that the values of the measurement process are conditionally independent given the latent Markov process. An intuitive way to present conditional independence properties between random variables is a *Bayes network* presented by a directed acyclic graph (DAG) (Pearl, 1988; Bishop, 2006). A Bayes network presentation of a discrete-time SSM is given in Figure 2.

The value $\mathbf{x}_k \in \mathcal{X} \equiv \mathbb{R}^{d_x}$ of the dynamic process at time t_k is called the *state* at time t_k. As explained in the introduction, the state summarizes as much information about the dynamic process as is needed to formulate the dynamic model introduced

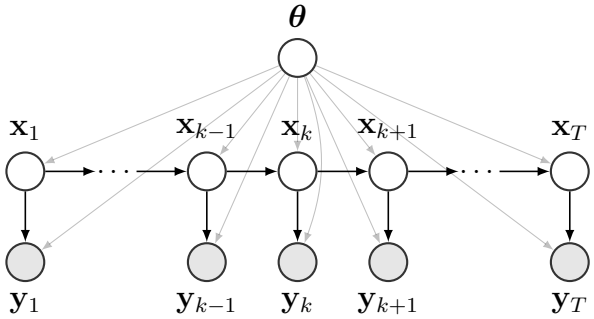

Figure 2: A discrete-time state space model as a graphical model presented with a directed acyclic graph. Each node represents a random variable and arrows present dependence. The hidden variables $\mathbf{x}_k$, meaning the states, form a Markov chain and each state has a corresponding measurement $\mathbf{y}_k$, which is oberved. Given the states, the measurements are independent. Both the states and the measurements depend on the parameter $\boldsymbol{\theta}$.

below. For the measurements we define $\mathbf{y}_k \in \mathcal{Y} \equiv \mathbb{R}^{d_y}$. As depicted in Figure 2, we assume that the joint *probability density function* (PDF, will be used interchangeably with *density* and *distribution*) of $\mathbf{x}_{0:T}$ and $\mathbf{y}_{0:T}$ is conditional on a set of parameters $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^{d_\theta}$.

Taking into account the Markov property

$$p(\mathbf{x}_k \,|\, \mathbf{x}_{1:k-1}, \boldsymbol{\theta}) = p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}, \boldsymbol{\theta}) \tag{1}$$

of the dynamic process and the conditional independence property

$$p(\mathbf{y}_k \,|\, \mathbf{x}_{1:k}, \mathbf{y}_{1:k-1}, \boldsymbol{\theta}) = p(\mathbf{y}_k \,|\, \mathbf{x}_k, \boldsymbol{\theta}) \tag{2}$$

of the measurement process, the joint density of states and measurements factorises as

$$p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T} \,|\, \boldsymbol{\theta}) = p(\mathbf{x}_0 \,|\, \boldsymbol{\theta}) \prod_{k=1}^{T} p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}, \boldsymbol{\theta}) \prod_{k=0}^{T} p(\mathbf{y}_k \,|\, \mathbf{x}_k, \boldsymbol{\theta}). \tag{3}$$

Thus in order to describe a SSM one needs to specify three distributions:

Prior distribution $p(\mathbf{x}_0 \,|\, \boldsymbol{\theta})$ is the distribution assumed for the state prior to observing any measurements. The sensitivity of the marginal posterio distribution to the prior depends on the amount of data (the more data the less sensitivity).

Dynamic model $p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}, \boldsymbol{\theta})$ dictates the time evolution of the states.

Measurement model $p(\mathbf{y}_k \mid \mathbf{x}_k, \boldsymbol{\theta})$ models how the observations depend on the state and the statistics of the noise.

In this thesis it is assumed that the parametric form of these distributions is known for example by physical modeling (Ljung & Glad, 1994). Regarding the notation, we will overload $p(\cdot \mid \cdot)$ as a generic probability density function specified by its arguments. Also the difference between random variables and their realizations is suppressed.

Traditionally SSMs are specified as a pair of equations specifying the dynamic and measurement models. In great generality, discrete-time SSMs can be described by the following dynamic and measurement equations

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{q}_{k-1}, \boldsymbol{\theta}) \tag{4a}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{r}_k, \boldsymbol{\theta}). \tag{4b}$$

Here the stochasticity is separated into the noise processes $\mathbf{q}$ and $\mathbf{r}$ which are usually assumed to be zero mean, white and independent of each other. We will restrict ourselves to the case of zero mean, white and additive Gaussian noise. Furthermore, the dynamic, measurement and both noise processes will be assumed *stationary*. This means that $\mathbf{f}_k$ and $\mathbf{h}_k$ and the PDFs of $\mathbf{q}_{k-1}$ and $\mathbf{r}_k$ will be independent of k. Thus the SSMs considered in this thesis are of the form

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \boldsymbol{\theta}) + \mathbf{q}_{k-1}, \qquad \mathbf{q}_{k-1} \sim \mathrm{N}\Big(\mathbf{0}, \mathbf{Q}(\boldsymbol{\theta})\Big) \tag{5a}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \boldsymbol{\theta}) + \mathbf{r}_k, \qquad \mathbf{r}_k \sim \mathrm{N}\Big(\mathbf{0}, \mathbf{R}(\boldsymbol{\theta})\Big) \tag{5b}$$

$$\mathbf{x}_0 \sim \mathrm{N}\Big(\boldsymbol{\mu}_0(\boldsymbol{\theta}), \boldsymbol{\Sigma}_0(\boldsymbol{\theta})\Big). \tag{5c}$$

Regarding the Gaussian probability distribution, suppose $\mathbf{x}$ is normally distributed with mean $\mathbf{m}$ and covariance matrix $\mathbf{P}$. We will then use the notation $\mathbf{x} \sim \mathrm{N}(\mathbf{m}, \mathbf{P})$ for "distributed as", whereas "distribution of" is denoted as $p(\mathbf{x}) = \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P})$, where the Gaussian probability density function is

$$\mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P}) \equiv \det(2\pi\mathbf{P})^{-1/2} \exp\Big(-1/2\,(\mathbf{x}-\mathbf{m})^{\mathsf{T}}\mathbf{P}^{-1}(\mathbf{x}-\mathbf{m})\Big). \tag{6}$$

Clearly the mappings $\mathbf{f} : \mathcal{X} \to \mathcal{X}$ and $\mathbf{h} : \mathcal{X} \to \mathcal{Y}$ in Equation (5) specify the means

of the dynamic and the measurement models:

$$p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}, \boldsymbol{\theta}) = \mathrm{N}\Big(\mathbf{x}_k \,\Big|\, \mathbf{f}(\mathbf{x}_{k-1}, \boldsymbol{\theta}), \mathbf{Q}(\boldsymbol{\theta})\Big) \tag{7a}$$

$$p(\mathbf{y}_k \,|\, \mathbf{x}_k, \boldsymbol{\theta}) = \mathrm{N}\Big(\mathbf{y}_k \,\Big|\, \mathbf{h}(\mathbf{x}_k, \boldsymbol{\theta}), \mathbf{R}(\boldsymbol{\theta})\Big). \tag{7b}$$

Going further, for the sake of notational clarity, we will sometimes make the dependence on $\boldsymbol{\theta}$ implicit and use the shorthand notation

$$\begin{aligned} \mathbf{f}_{k-1} &\equiv \mathbf{f}(\mathbf{x}_{k-1}, \boldsymbol{\theta}), \quad & \mathbf{h}_k &\equiv \mathbf{h}(\mathbf{x}_k, \boldsymbol{\theta}) \\ \mathbf{Q} &\equiv \mathbf{Q}(\boldsymbol{\theta}), & \mathbf{R} &\equiv \mathbf{R}(\boldsymbol{\theta}) \\ \boldsymbol{\mu}_0 &\equiv \boldsymbol{\mu}_0(\boldsymbol{\theta}), & \boldsymbol{\Sigma}_0 &\equiv \boldsymbol{\Sigma}_0(\boldsymbol{\theta}). \end{aligned} \tag{8}$$

2.2 Bayesian optimal filtering and smoothing

State inference can be divided into subcategories based on the temporal relationship between the state and the observations (see, e.g., Särkkä, 2006; Anderson & Moore, 1979):

Predictive distribution $p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k-1}, \boldsymbol{\theta})$ is the predicted distribution of the state in the next timestep (or more generally at timestep $k + h$, where $h > 0$) given the previous measurements.

Filtering distribution $p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k}, \boldsymbol{\theta})$ is the marginal posterior distribution of any state $\mathbf{x}_k$ given the measurements up to and including $\mathbf{y}_k$.

Smoothing distribution $p(\mathbf{x}_k \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta})$ is the marginal posterior distribution of any state $\mathbf{x}_k$, $k = 1, \ldots, T$, given the measurements up to and including $\mathbf{y}_T$.

Predictive distribution

Let us then derive a recursive formulation for computing the filtering distribution at time k. Let $p(\mathbf{x}_{k-1} \,|\, \mathbf{y}_{1:k-1})$ be the filtering distribution of the previous step. Then

$$\begin{aligned} p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k-1}, \boldsymbol{\theta}) &= \int p(\mathbf{x}_k, \mathbf{x}_{k-1} \,|\, \mathbf{y}_{0:k-1}, \boldsymbol{\theta}) \,\mathrm{d}\mathbf{x}_{k-1} \\ &= \int p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \,|\, \mathbf{y}_{0:k-1}, \boldsymbol{\theta}) \,\mathrm{d}\mathbf{x}_{k-1}, \end{aligned} \tag{9}$$

which is known as the *Chapman-Kolmogorov equation* (see,e.g., Särkkä, 2006). In this thesis the predictive distributions will be Gaussian or approximated with a

Gaussian

$$p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k-1}, \boldsymbol{\theta}) \approx \mathrm{N}\Big(\mathbf{x}_k \,\Big|\, \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}\Big). \tag{10}$$

Filtering distribution

Incorporating the newest measurement can be achieved with the Bayes' rule (see, e.g., Gelman et al., 2004)

$$\begin{aligned} \underbrace{p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k}, \boldsymbol{\theta})}_{\text{posterior}} &= \frac{\overbrace{p(\mathbf{y}_k \,|\, \mathbf{x}_k, \boldsymbol{\theta})}^{\text{likelihood}} \overbrace{p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k-1}, \boldsymbol{\theta})}^{\text{prior}}}{\underbrace{p(\mathbf{y}_k \,|\, \mathbf{y}_{0:k-1}, \boldsymbol{\theta})}_{\text{normalization constant}}} \\ &= \frac{p(\mathbf{y}_k \,|\, \mathbf{x}_k) p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k-1})}{\int p(\mathbf{y}_k \,|\, \mathbf{x}_k) p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k-1}) \,\mathrm{d}\mathbf{x}_k}, \end{aligned} \tag{11}$$

which is called the measurement update equation. In this thesis the filtering distributions will be Gaussian or approximated with a Gaussian

$$p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k}, \boldsymbol{\theta}) \approx \mathrm{N}\Big(\mathbf{x}_k \,\Big|\, \mathbf{m}_{k|k}, \mathbf{P}_{k|k}\Big). \tag{12}$$

Smoothing distribution

The smoothing distributions can also be computed recursively by assuming that the filtering distributions and the smoothing distribution $p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:T})$ of the "previous" step are available. Since

$$\begin{aligned} p(\mathbf{x}_k \,|\, \mathbf{x}_{k+1}, \mathbf{y}_{0:T}, \boldsymbol{\theta}) &= p(\mathbf{x}_k \,|\, \mathbf{x}_{k+1}, \mathbf{y}_{0:k}, \boldsymbol{\theta}) \\ &= \frac{p(\mathbf{x}_k, \mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:k}, \boldsymbol{\theta})}{p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:k}, \boldsymbol{\theta})} \\ &= \frac{p(\mathbf{x}_{k+1} \,|\, \mathbf{x}_k, \boldsymbol{\theta}) p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k}, \boldsymbol{\theta})}{p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:k}, \boldsymbol{\theta})} \end{aligned}$$

we get

$$p(\mathbf{x}_k, \mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}) = \overbrace{p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k}, \boldsymbol{\theta})}^{\text{filtering}} \frac{\overbrace{p(\mathbf{x}_{k+1} \,|\, \mathbf{x}_k, \boldsymbol{\theta})}^{\text{dynamic}} p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta})}{\underbrace{p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:k}, \boldsymbol{\theta})}_{\text{predictive}}}, \tag{13}$$

so that the marginal is given by

$$p(\mathbf{x}_k \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}) = p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k}, \boldsymbol{\theta}) \int \left[\frac{p(\mathbf{x}_{k+1} \,|\, \mathbf{x}_k, \boldsymbol{\theta}) p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta})}{p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:k}, \boldsymbol{\theta})} \right] \mathrm{d}\mathbf{x}_{k+1}, \tag{14}$$

where $p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:k})$ can be computed by Equation (9). In this thesis the smoothing distributions will be Gaussian or approximated with a Gaussian

$$p(\mathbf{x}_k \,|\, \mathbf{y}_{0:T}) \approx \mathrm{N}\Big(\mathbf{x}_k \,\Big|\, \mathbf{m}_{k|T}, \mathbf{P}_{k|T}\Big). \tag{15}$$

Marginal likelihood

An important quantity concerning parameter estimation is the marginal likelihood $p(\mathbf{y}_{0:T} \,|\, \boldsymbol{\theta})$. If we're able to compute the distributions

$$p(\mathbf{y}_k \,|\, \mathbf{y}_{0:k-1}, \boldsymbol{\theta}) = \int p(\mathbf{y}_k \,|\, \mathbf{x}_k, \boldsymbol{\theta}) p(\mathbf{x}_k \,|\, \mathbf{y}_{0:k-1}, \boldsymbol{\theta}) \, \mathrm{d}\mathbf{x}_k, \tag{16}$$

which we recognize as the "normalization constant" in (11), then by repeatedly applying the definition of conditional probability we find that the marginal likelihood can be computed from

$$p(\mathbf{y}_{0:T} \,|\, \boldsymbol{\theta}) = p(\mathbf{y}_0 \,|\, \boldsymbol{\theta}) \prod_{k=1}^{T} p(\mathbf{y}_k \,|\, \mathbf{y}_{0:k-1}, \boldsymbol{\theta}). \tag{17}$$

Since (16) is needed for the filtering distributions, the marginal likelihood, or an approximation to it, can be easily computed with the chosen filtering algorithm. Equation (17) is sometimes known as the *prediction error decomposition* (Harvey, 1990).

3 State estimation

In this section we are concerned with finding the, exact if possible and approximate otherwise, filtering and smoothing distributions of Equations (11) and (14). In fact, since it is needed in parameter estimation, we will focus on the somewhat more general problem of finding the cross-timestep joint densities. In state estimation it is assumed that the parameter $\boldsymbol{\theta}$ is given. Thus throughout this section we will, in general, suppress the dependence on $\boldsymbol{\theta}$ and use the shorthands specified in (8).

3.1 Linear-Gaussian State Space Models

A linear map $Q : \mathcal{A} \to \mathcal{B}$ satisfies the equation

$$Q(\alpha\mathbf{a} + \beta\mathbf{b}) = \alpha Q(\mathbf{a}) + \beta Q(\mathbf{b}), \quad \forall\ \mathbf{a}, \mathbf{b} \in \mathcal{A}\ \ \&\ \ \alpha, \beta \in \mathbb{R}. \tag{18}$$

Since linear maps can be described by matrices, stationary linear-Gaussian SSMs are described by the subset of SSMs of the form (5) where

$$\mathbf{f}(\mathbf{x}_{k-1}, \boldsymbol{\theta}) = \mathbf{A}(\boldsymbol{\theta})\mathbf{x}_{k-1} \tag{19}$$

$$\mathbf{h}(\mathbf{x}_k, \boldsymbol{\theta}) = \mathbf{H}(\boldsymbol{\theta})\mathbf{x}_k. \tag{20}$$

The $d_x \times d_x$ matrix $\mathbf{A} \equiv \mathbf{A}(\boldsymbol{\theta})$ is called the *transition matrix* and the $d_y \times d_x$ matrix $\mathbf{H} \equiv \mathbf{H}(\boldsymbol{\theta})$ the *measurement matrix*. These linear-Gaussian SSMs, equivalently known as *linear dynamical systems* (Bishop, 2006), are one of the few cases where computing the *exact* predictive, filtering and smoothing distributions is tractable (see, e.g., Särkkä, 2006). As will be seen, all of the aforementioned distributions stay Gaussian.

3.1.1 Kalman filter

The Kalman filter is the best known filter, first presented in the seminal article of Kalman (1960). It provides the closed form solution to computing the predictive and filtering distributions of Equations (9) and (11). With the help of Lemmas A.1 and A.2, deriving the Kalman filter equations is quite straightforward (Särkkä, 2006). The resulting recursions are (Jazwinski, 1970; Grewal & Andrews, 2008):

Predict:

$$\mathbf{m}_{k|k-1} = \mathbf{A}\mathbf{m}_{k-1|k-1} \tag{21a}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^{\mathsf{T}} + \mathbf{Q} \tag{21b}$$

Update:

$$\mathbf{v}_k = \mathbf{y}_k - \mathbf{H}\mathbf{m}_{k|k-1} \tag{21c}$$
$$\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^{\mathsf{T}} + \mathbf{R} \tag{21d}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^{\mathsf{T}}\mathbf{S}_k^{-1} \tag{21e}$$
$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k\mathbf{v}_k \tag{21f}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^{\mathsf{T}}. \tag{21g}$$

This includes the sufficient statistics for the T joint distributions

$$p(\mathbf{x}_k, \mathbf{y}_k \,|\, \mathbf{y}_{1:k-1}, \boldsymbol{\theta}) = \mathrm{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} \middle| \begin{bmatrix} \mathbf{m}_{k|k-1} \\ \mathbf{H}\mathbf{m}_{k|k-1} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{k|k-1} & \mathbf{P}_{k|k-1}\mathbf{H}^{\mathsf{T}} \\ \mathbf{H}\mathbf{P}_{k|k-1} & \mathbf{S}_k \end{bmatrix} \right). \tag{22}$$

3.1.2 Rauch–Tung–Striebel Smoother

Once the filtering distributions are obtained going *forward* in time, the joint smoothing dustributions (13) can be computed going *backwards* in time. In this computing order sense, the last filtering distribution is the first smoothing distribution. In the linear-Guassian case, the Rauch–Tung–Striebel (RTS) smoother gives the statistics $\mathbf{m}_{k|T}$ and $\mathbf{P}_{k|T}$ (Jazwinski, 1970; Rauch, Tung, & Striebel, 1965) in Equation (15). We will use a version that gives the joint distribution of the states across a timestep, since the cross-timestep covariance will be needed in the parameter estimation phase. Assuming now that all the predictive and filtering distributions, that is $\mathrm{N}\left(\mathbf{x}_{k+1} \,\middle|\, \mathbf{m}_{k+1|k}, \mathbf{P}_{k+1|k}\right)$ and $\mathrm{N}\left(\mathbf{x}_k \,\middle|\, \mathbf{m}_{k|k}, \mathbf{P}_{k|k}\right)$ respectively, are available, the RTS recursions can be written as

$$\mathbf{J}_k = \mathbf{P}_{k|k}\mathbf{A}^{\mathsf{T}}\mathbf{P}_{k|k+1}^{-1} \tag{23a}$$
$$\mathbf{m}_{k|T} = \mathbf{m}_{k|k} + \mathbf{J}_k\left(\mathbf{m}_{k+1|T} - \mathbf{m}_{k+1|k}\right) \tag{23b}$$
$$\mathbf{P}_{k|T} = \mathbf{P}_{k|k} + \mathbf{J}_k\left(\mathbf{P}_{k+1|T} - \mathbf{P}_{k+1|k}\right)\mathbf{J}_k^{\mathsf{T}}. \tag{23c}$$

This includes the sufficient statistics for the T joint distributions

$$p(\mathbf{x}_k, \mathbf{x}_{k-1} \mid \mathbf{Y}, \boldsymbol{\theta}) = \mathrm{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{bmatrix} \middle| \begin{bmatrix} \mathbf{m}_{k|T} \\ \mathbf{m}_{k-1|T} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{k|T} & \mathbf{P}_{k|T}\mathbf{J}_k^\mathsf{T} \\ \mathbf{J}_k\mathbf{P}_{k|T} & \mathbf{P}_{k-1|T} \end{bmatrix} \right). \tag{24}$$

3.2 Nonlinear-Gaussian SSMs

In the nonlinear case at least one of the mappings $\mathbf{f}$ and $\mathbf{h}$ in (5) is nonlinear (in $\mathbf{x}$). Unfortunately in this case computing the filtering distributions in closed form becomes intractable and one has to resort to some sort of approximations. We can divide these approximate filtering (and smoothing) solutions into two categories (see, e.g., Arasaratnam & Haykin, 2009):

i) *Local approaches* assume the parametric form of the posterior distributions (9), (11) and (14) *a priori*. These methods are analytically inexact but less computationally demanding. This is the category that we will be concerned with in this thesis.

ii) *Global approaches* require the use of particle filtering, also known as *sequential Monte Carlo* (SMC), methods, which are *simulation* based.

The number of different methods in the first category is substantial, but a large proportion can be analyzed under the framework of *Gaussian filtering* (or assumed density filtering with a Gaussian assumption). As implied by the name, these methods work by restricting the form of the posterior density to be Gaussian a priori. This way one is again able to perform (approximate) filtering and smoothing by only propagating the first two moments, which makes the local approaches computationally efficient. As will be shown later, the specific Gaussian filtering methods only differ in their chosen numerical integration methods.

The global approaches are certainly appealing in not placing any restrictions on the form of the posterior distribution. Particle filtering has been enjoying widespred interest since the introduction in Gordon, Salmond, and Smith (1993) (see also Cappé, Godsill, & Moulines, 2007; Kantas, Doucet, & Singh, 2009; Cappé et al., 2005). However they are Monte Carlo methods, the use of which usually requires tuning and convergence monitoring. The most obvious downside compared to the local methods are their increased computational requirements.

3.2.1 Gaussian filtering and smoothing

One approach to forming Gaussian approximations is to assume a Gaussian probability distribution a priori (Kushner, 1967; Ito, 2000; Y. Wu, Hu, Wu, & Hu, 2006; Särkkä & Hartikainen, 2010). Since a Gaussian distribution is defined by its first two moments, a moment matched approximation can be obtained if the first two moments of the actual probability distribution can be computed (Ito, 2000; Särkkä, 2006). As will be seen, computing these Gaussian approximations reduces to the problem of computing multidimensional moment integrals of the form *nonlinear function* $\times$ *Gaussian*.

We shall next derive the general form of the three moment integrals and then show how they can be applied in the specific case of approximating the joint smoothing distribution of Equation (13). Suppose now that

$$\begin{aligned} p(\mathbf{x}) &= \mathrm{N}(\mathbf{x} \,|\, \mathbf{m}, \mathbf{P}), \\ p(\mathbf{y} \,|\, \mathbf{x}) &= \mathrm{N}(\mathbf{y} \,|\, \mathbf{f}(\mathbf{x}), \mathbf{R}). \end{aligned}$$

Then $p(\mathbf{x}, \mathbf{y})$ is Gaussian only if $\mathbf{f}(\mathbf{x})$ is a linear map (with a possible affine constant). Assuming that's not the case, let us denote a Gaussian approximation to $p(\mathbf{x}, \mathbf{y})$ with

$$p\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}\right) \approx \mathrm{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x\\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy}\\ \boldsymbol{\Sigma}_{xy}^{\mathsf{T}} & \boldsymbol{\Sigma}_{yy}\end{bmatrix}\right).$$

Then according to Lemmas A.1 and A.2, we have to have

$$\begin{aligned} \boldsymbol{\mu}_x &= \mathbf{m} \\ \boldsymbol{\Sigma}_{xx} &= \mathbf{P} \\ \boldsymbol{\mu}_y &= \int \mathbf{f}(\mathbf{x}) \mathrm{N}(\mathbf{x} \,|\, \mathbf{m}, \mathbf{P}) \,\mathrm{d}\mathbf{x} && (25) \\ \boldsymbol{\Sigma}_{yy} &= \int \Big(\mathbf{f}(\mathbf{x}) - \boldsymbol{\mu}_y\Big)\Big(\mathbf{f}(\mathbf{x}) - \boldsymbol{\mu}_y\Big)^{\mathsf{T}} \mathrm{N}(\mathbf{x} \,|\, \mathbf{m}, \mathbf{P}) \,\mathrm{d}\mathbf{x} + \mathbf{R} && (26) \\ \boldsymbol{\Sigma}_{xy} &= \int \Big(\mathbf{x} - \mathbf{m}\Big)\Big(\mathbf{f}(\mathbf{x}) - \boldsymbol{\mu}_y\Big)^{\mathsf{T}} \mathrm{N}(\mathbf{x} \,|\, \mathbf{m}, \mathbf{P}) \,\mathrm{d}\mathbf{x} && (27) \end{aligned}$$

Prediction step

Since the Gaussian approximation to (13) will be calculated by forward (filtering) and backward (smoothing) recursions, let us assume that we already have available

the filtering distribution of the previous step

$$p(\mathbf{x}_{k-1} \,|\, \mathbf{y}_{1:k-1}) \approx \mathrm{N}\Big(\mathbf{x}_{k-1} \,\Big|\, \mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}\Big). \tag{28}$$

Then

$$\begin{aligned} p(\mathbf{x}_{k-1}, \mathbf{x}_k \,|\, \mathbf{y}_{1:k-1}) &\approx \mathrm{N}\Big(\mathbf{x}_{k-1} \,\Big|\, \mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}\Big)\mathrm{N}(\mathbf{x}_k \,|\, \mathbf{f}_{k-1}, \mathbf{Q}) \\ &\approx \mathrm{N}\left(\begin{bmatrix}\mathbf{x}_{k-1}\\ \mathbf{x}_k\end{bmatrix} \,\middle|\, \begin{bmatrix}\mathbf{m}_{k-1|k-1}\\ \mathbf{m}_{k|k-1}\end{bmatrix}, \begin{bmatrix}\mathbf{P}_{k-1|k-1} & \mathbf{P}_{k-1,k}\\ \mathbf{P}_{k-1,k}^{\mathsf{T}} & \mathbf{P}_{k|k-1}\end{bmatrix}\right), \end{aligned} \tag{29}$$

where by application of Equations (25), (26) and (27)

$$\mathbf{m}_{k|k-1} = \int \mathbf{f}_{k-1}\mathrm{N}\Big(\mathbf{x}_{k-1} \,\Big|\, \mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}\Big)\,\mathrm{d}\mathbf{x}_{k-1} \tag{30}$$

$$\begin{aligned} \mathbf{P}_{k|k-1} = \int &\Big(\mathbf{f}_{k-1} - \mathbf{m}_{k|k-1}\Big)\Big(\mathbf{f}_{k-1} - \mathbf{m}_{k|k-1}\Big)^{\mathsf{T}} \\ &\times \mathrm{N}\Big(\mathbf{x}_{k-1} \,\Big|\, \mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}\Big)\,\mathrm{d}\mathbf{x}_{k-1} + \mathbf{Q} \end{aligned} \tag{31}$$

$$\begin{aligned} \mathbf{P}_{k-1,k} = \int &\Big(\mathbf{x}_{k-1} - \mathbf{m}_{k-1|k-1}\Big)\Big(\mathbf{f}_{k-1} - \mathbf{m}_{k|k-1}\Big)^{\mathsf{T}} \\ &\times \mathrm{N}(\mathbf{x}_{k-1}, \mathbf{m}_{k-1|T})\,\mathrm{d}\mathbf{x}_{k-1} \end{aligned}. \tag{32}$$

Update step

For the update step we first approximate

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{y}_k \,|\, \mathbf{y}_{1:k-1}) &\approx \mathrm{N}(\mathbf{y}_k \,|\, \mathbf{h}_k, \mathbf{R})\mathrm{N}\Big(\mathbf{x}_k \,\Big|\, \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}\Big) \\ &\approx \mathrm{N}\left(\begin{bmatrix}\mathbf{x}_k\\ \mathbf{y}_k\end{bmatrix} \,\middle|\, \begin{bmatrix}\mathbf{m}_{k|k-1}\\ \boldsymbol{\mu}_k\end{bmatrix}, \begin{bmatrix}\mathbf{P}_{k|k-1} & \mathbf{C}_k\\ \mathbf{C}_k^{\mathsf{T}} & \mathbf{S}_k\end{bmatrix}\right). \end{aligned} \tag{33}$$

Applying Equations (25), (26) and (27) again, we get

$$\boldsymbol{\mu}_k = \int \mathbf{h}_k\mathrm{N}\Big(\mathbf{x}_k \,\Big|\, \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}\Big)\,\mathrm{d}\mathbf{x}_k \tag{34}$$

$$\mathbf{S}_k = \int \Big(\mathbf{h}_k - \boldsymbol{\mu}_k\Big)\Big(\mathbf{h}_k - \boldsymbol{\mu}_k\Big)^{\mathsf{T}}\mathrm{N}\Big(\mathbf{x}_k \,\Big|\, \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}\Big)\,\mathrm{d}\mathbf{x}_k + \mathbf{R} \tag{35}$$

$$\mathbf{C}_k = \int \Big(\mathbf{x}_k - \mathbf{m}_{k|k-1}\Big)\Big(\mathbf{h}_k - \boldsymbol{\mu}_k\Big)^{\mathsf{T}}\mathrm{N}\Big(\mathbf{x}_k \,\Big|\, \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}\Big)\,\mathrm{d}\mathbf{x}_k. \tag{36}$$

The approximation to the filtering distribution $p(\mathbf{x}_k \,|\, \mathbf{y}_{1:k}) \approx \mathrm{N}\Big(\mathbf{x}_k \,\Big|\, \mathbf{m}_{k|k}, \mathbf{P}_{k|k}\Big)$ is then given by applying Lemma A.2 to (33). Analogously to the update equations

of the Kalman filter (21), we get

$$\mathbf{v}_k = \mathbf{y}_k - \boldsymbol{\mu}_k \tag{37a}$$
$$\mathbf{K}_k = \mathbf{C}_k \mathbf{S}_k^{-1} \tag{37b}$$
$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k \mathbf{v}_k \tag{37c}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^\mathsf{T}. \tag{37d}$$

Smoothing step

Let us write down the approximation to a conditional distribution that is easily derived from Equation (29), namely (note the change in indexing):

$$p(\mathbf{x}_k \,|\, \mathbf{x}_{k+1}, \mathbf{y}_{1:T}) = p(\mathbf{x}_k \,|\, \mathbf{x}_{k+1}, \mathbf{y}_{1:k}) \approx \mathrm{N}\Big(\mathbf{x}_k \,\Big|\, \mathbf{m}'_k, \mathbf{P}'_k\Big), \tag{38}$$

where

$$\begin{aligned}
\mathbf{m}'_k &= \mathbf{m}_{k|k} + \mathbf{G}_k\Big(\mathbf{x}_{k+1} - \mathbf{m}_{k+1|k}\Big) \\
\mathbf{P}'_k &= \mathbf{P}_{k|k} - \mathbf{G}_k \mathbf{P}_{k+1|k} \mathbf{G}_k^\mathsf{T} \\
\mathbf{G}_k &= \mathbf{P}_{k,k+1} \mathbf{P}_{k+1|k}^{-1}.
\end{aligned}$$

At this point we have derived all the components needed to compute (13). As pointed out previously, the last (T:th), filtering distribution is also the "first" smoothing distribution, and smoothing recursions then advance backwards in time. Let us assume that the smoothing distribution of the previous step, $p(\mathbf{x}_{k+1} \,|\, \mathbf{y}_{1:T})$, is available. Then by applying Lemma A.1 we have

$$p(\mathbf{x}_k, \mathbf{x}_{k+1} \,|\, \mathbf{y}_{0:T}) \approx \mathrm{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{bmatrix} \,\middle|\, \begin{bmatrix} \mathbf{m}_{k|T} \\ \mathbf{m}_{k+1|T} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{k|T} & \mathbf{D}_k \\ \mathbf{D}_k^\mathsf{T} & \mathbf{P}_{k+1|T} \end{bmatrix}\right), \tag{39}$$

where

$$\begin{aligned}
\mathbf{D}_k &= \mathbf{G}_k \mathbf{P}_{k+1|T} \\
\mathbf{m}_{k|T} &= \mathbf{m}_{k|k} + \mathbf{G}_k \left(\mathbf{m}_{k+1|T} - \mathbf{m}_{k+1|k}\right) \\
\mathbf{P}_{k|T} &= \mathbf{P}_{k|k} + \mathbf{G}_k \left(\mathbf{P}_{k+1|T} - \mathbf{P}_{k+1|k}\right) \mathbf{G}_k^\mathsf{T}.
\end{aligned}$$

What we have now established is that a Gaussian assumed density approximation to the joint smoothing distributionof Equation (13) is transformed into solving six

multidimensional integrals of form *nonlinear function* × *Gaussian*, namely the ones in (30), (31), (32), (34), (35) and (36). Notably, the smoothing distribution approximations can be computed without further integrations.

3.2.2 Numerical integration approach

We will now discuss the topic of numerically solving Gaussian expectation integrals of the form

$$\langle \boldsymbol{\kappa}(\mathbf{x}) \rangle \equiv \int_{\mathcal{X}} \boldsymbol{\kappa}(\mathbf{x}) \mathrm{N}(\mathbf{x} \,|\, \mathbf{m}, \mathbf{P}) \,\mathrm{d}\mathbf{x}, \tag{40}$$

where it is assumed that

$$\begin{aligned} \mathcal{X} &= \mathbb{R}^{d_x} \\ \mathbf{x} &\sim \mathrm{N}(\mathbf{m}, \mathbf{P}) \\ \int |\boldsymbol{\kappa}(\mathbf{x}) \mathrm{N}(\mathbf{x} \,|\, \mathbf{m}, \mathbf{P})| \,\mathrm{d}\mathbf{x} &< \infty. \end{aligned}$$

As explained in Y. Wu et al. (2006), the approaches to solving (40) can be justifiably divided into three categories:

i) product rules
ii) rules exact for monomials
iii) integrand approximations.

Recognizing that the chosen numerical integration method is the principal differentiator provides a common framework for analyzing the properties of the numerous Gaussian filters and smoothers (Särkkä & Hartikainen, 2010; Särkkä & Sarmavuori, 2013; Ito, 2000; Y. Wu et al., 2006). Furthermore the first two categories differ only in their approach to multidimensional integrals, so that the main difference between the categories can be described as applying an integration formula known to be exact for certain class of integrands or approximating the integrand and integrating the approximation exactly. Since truncated Taylor series approximations are often used in the latter case, an important distinction is that the former does not require computation of Jacobians or higher order differentials.

Since there exists many efficient integration rules defined on the unidimensional line, a natural idea is to extend these to the hypercube by iterated integrals. This is exactly the basic premise of the product rules. The most efficient polynomial interpolation type of rules in one dimension are known as *Gauss' quadrature* rules and

the subset for Gaussian weighted integrals are the *Gauss-Hermite* quadrature rules. Quadrature is a term referring to unidimensional numerical integration, whereas *cubature* is the generalization to higher dimensions. A common form for cubature rules for Gaussian expectation integrals is

$$\int_{\mathcal{X}} \boldsymbol{\kappa}(\mathbf{x}) \mathrm{N}(\mathbf{x} \,|\, \mathbf{m}, \mathbf{P}) \,\mathrm{d}\mathbf{x} \approx \sum_{i=1}^{m} w_i \boldsymbol{\kappa}(\mathbf{u}_i) = \sum_{i=1}^{m} w_i \boldsymbol{\kappa}\Big(\mathbf{m} + \sqrt{\mathbf{P}} \boldsymbol{\varepsilon}^{(i)}\Big), \tag{41}$$

where the points of evaluation $\{\mathbf{u}_i\}_{i=1}^m$ are called the *sigma points* (or *abscissas* or just points) and w_i are the weights. The sigma points can be obtained from the *unit* sigma points $\left\{\boldsymbol{\varepsilon}^{(i)}\right\}_{i=1}^m$ by translating with the mean and scaling by the Cholesky decomposition of the covariance matrix, $\mathbf{P} = \sqrt{\mathbf{P}}\sqrt{\mathbf{P}}^{\mathsf{T}}$. This means that to specify any cubature rule of the form (41), it suffices to specify it for the case with zero mean and unit covariance matrix

$$\int_{\mathcal{X}} \boldsymbol{\kappa}(\mathbf{x}) \mathrm{N}(\mathbf{x} \,|\, \mathbf{0}, \mathbf{I}) \,\mathrm{d}\mathbf{x} \approx \sum_{i=1}^{m} w_i \boldsymbol{\kappa}\Big(\boldsymbol{\varepsilon}^{(i)}\Big). \tag{42}$$

As was to be expected from the iterated integration approach, the problem with product rules is the exponential increase in the number of sigma points with the number of dimensions, also known as the *curse of dimensionality*. Thus if the unidimensional rule has m sigma points (and thus m integrand evaluations), then the d dimensional product rule has m^d sigma points. The Gaussian filter based on Gauss-Hermite product rules is known simply as the Gauss-Hermite Kalman filter (GHKF, sometimes shortened also GKF or QKF, for quadrature Kalman filter) (Ito, 2000). The number of sigma points in the unidimensional rule is a parameter of GHKF.

More sophisticated cubature methods search for rules exact for *monomials* $\prod_{j=1}^{d_x} x_j^{e_j}$, where $\mathbf{x} = \left[x_1, \ldots, x_{d_x}\right]^{\mathsf{T}}$. The *degree* of the monomial is defined as $\sum_j e_j$ and a cubature rule is then said to have *precision* p, if it integrates exactly monomials up to degree p but not to degree $p+1$. Naturally since integration is a linear operation, a rule which is exact for a monomial up to order o is exact for multidimensional polynomials of order o. Unfortunately finding efficient rules exact for monomials is something of an art, since even the least possible number of points required for given precision and dimension is in many instances unknown. Nevertheless, following the work in Y. Wu et al. (2006), in Arasaratnam and Haykin (2009, 2011) a filter and a corresponding smoother are presented, based on a third degree cubature rule. The theoretical lower bound in points for a third degree rule is $2\,d_x$, which is met by the

rule used in the *Cubature Kalman Filter* (CKF) and the *Cubature Kalman Smoother* (CKS). Another notable nonlinear filter in this category is the *Unscented Kalman Filter* (UKF) (Julier & Uhlmann, 1997; Julier, Uhlmann, & Durrant-Whyte, 2000; Merwe, 2004), also based on a third degree rule. A corresponding smoother is derived in Särkkä (2008). An interesting recent development which could be considered to belong between the product rules and rules exact for monomials is presented in Jia, Xin, and Cheng (2012), where the method of *sparse-grid quadrature* is used to obtain yet another nonlinear filter belonging to the class of Gaussian filters.

The oldest and most well known nonlinear filter, belonging to the third category, is the *extended Kalman filter* (EKF) (see, e.g., Grewal & Andrews, 2008). It is based on forming local linear approximations to the dynamic and measurement models so that the standard linear Kalman filter equations can be used. An undesirable requirement of the EKF is that it requires computing the Jacobian matrices of $\mathbf{f}$ and $\mathbf{h}$.

3.2.3 Cubature Kalman Filter and Smoother

In this subsection we will present the Cubature Kalman Filter (CKF) and Cubature Kalman Smoother (CKS) algorithms (Arasaratnam & Haykin, 2011, 2009). We consider these algorithms in more detail than other Gaussian filters and smoothers, since they are applied in Section 5. The CKF results from applying a *3rd order spherical cubature approximation* to the integrals in Equations (30), (31), (32), (34), (35) and (36).

As stated earlier, the 3rd order spherical cubature approximation uses the minimal amount of sigma points for a 3rd order rule, $2\,d_x$. The unit sigma points in Equation (42) are given by

$$\begin{aligned} \boldsymbol{\varepsilon}^{(i)} &= \quad \sqrt{d_x}\mathbf{e}_i, && i = 1, \ldots, d_x \\ \boldsymbol{\varepsilon}^{(i)} &= -\sqrt{d_x}\mathbf{e}_{i-d_x}, && i = d_x + 1, \ldots, 2\,d_x, \end{aligned} \tag{43}$$

where the orthonormal basis vectors $\{\mathbf{e}_i\}_{i=1}^{d_x}$ form the canonical basis of $\mathbb{R}^{d_x}$. The weight is constant, $w_i \equiv w = \frac{1}{2\,d_x}$, so that approximation (41) can be written as

$$\int_{\mathcal{X}} \boldsymbol{\kappa}(\mathbf{x}) \mathrm{N}(\mathbf{x} \,|\, \mathbf{m}, \mathbf{P})\, \mathrm{d}\mathbf{x} \approx \frac{1}{2\,d_x} \sum_{i=1}^{2\,d_x} \boldsymbol{\kappa}\Big(\mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\varepsilon}^{(i)}\Big). \tag{44}$$

4 Parameter estimation

As mentioned in the introduction, it is usually the case that after constructing a SSM, the result is a family of models indexed by the static parameter $\boldsymbol{\theta}$. The ultimate interest might lie in estimating the states or the parameter or both. Be as it may, the two inference problems are intimately coupled and interest in the other requires the resolution of the other.

In general, parameter estimation techniques are divided into offline or *batch* methods and online or *recursive* methods (Cappé et al., 2007; Kantas et al., 2009). This is analogous to the difference between the filtering and smoothing problems in state estimation. We focus only on offline methods, where some sort of training or calibration data has been acquired beforehand.

A classic solution to the parameter estimation problem is to introduce an augmented and thus necessarily nonlinear SSM, where the parameters have been concatenated as part of the state. For static parameters, the part of the dynamic model corresponding to the parameters is set to identity. Classically an extended Kalman filter is then applied to approximate the probability distribution of the augmented state vector in the joint space of parameters and states. This approach is known as *joint EKF* and it has the virtue of being an online procedure (Wan & Nelson, 2001). It appears that the method has problems with convergence in some situations, which is understandable since when using the EKF, a Gaussian approximation is applied to the *joint* space of states and parameters.

A more recent method utilizing another form of augmented SSM is known as *iterated filtering* (Ionides, Bhadra, Atchadé, & King, 2011). It is an offline method, but only requires being able to sample from the dynamic model given the parameter and no gradient computations are required. The algorithm however introduces multiple parameters of its own and so might require some tuning (Kantas et al., 2009). Furthermore, it is designed to utilize the simulation based SMC methods mentioned briefly in Section 3.2. In the sequel, parameter estimation methods based on state augmentation will not be further considered.

4.1 Bayesian Estimation of Parameters

In the Bayesian sense the complete answer to the filtering and parameter estimation problems would be the *joint* posterior distribution of the states and the parameters

given the data

$$\begin{aligned} p(\mathbf{x}_{0:T}, \boldsymbol{\theta} \,|\, \mathbf{y}_{0:T}) &\propto p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T} \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta}) \\ &= p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}) p(\mathbf{y}_{0:T} \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta}). \end{aligned} \tag{45}$$

By defining the SSM in Equation (5), we have implicitly defined the "complete-data" likelihood $p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T} \,|\, \boldsymbol{\theta})$ (see Equation (3)). By introducing the *prior distribution*, $p(\boldsymbol{\theta})$, the components of (45) and thus the joint distribution posterior of states and parametes is defined. Recently, methods known as *Particle Markov chain Monte Carlo* (PMCMC) have emerged, which are able to sample from the joint distribution in (45) without knowledge of the normalization constant (Andrieu, Doucet, & Holenstein, 2010). This is achieved by combining particle filtering approximations to $p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta})$ with traditional Gibbs and Metropolis-Hastings sampling in a nontrivial way (Andrieu et al., 2010; Gelman et al., 2004).

4.1.1 Maximum a posteriori and maximum likelihood

In this thesis we would like to avoid Monte Carlo methods altogether. Thus instead of considering the problem of finding the posterior distribution of the parameter, we will pursue finding the mode of this distribution, that is, the *maximum a posteriori* (MAP) estimate $\boldsymbol{\theta}_{\text{MAP}}$. The MAP estimate is not necessarily unique, but let us assume for the moment that the posterior distribution in fact has a unique maximum. Since the logarithm is a strictly monotonic function, maximizing a function is the same as maximizing its logarithm. Since $\mathbf{y}_{0:T}$ is observed, let us denote the log marginal likelihood with

$$\ell\Big(\boldsymbol{\theta}\Big) \equiv \log p(\mathbf{y}_{0:T} \,|\, \boldsymbol{\theta}).$$

The MAP estimate of $\boldsymbol{\theta}$ is then defined as

$$\begin{aligned} \boldsymbol{\theta}_{\text{MAP}} &\equiv \arg\max_{\boldsymbol{\theta}} \Big[\log p(\boldsymbol{\theta} \,|\, \mathbf{y}_{0:T})\Big] \\ &= \arg\max_{\boldsymbol{\theta}} \Big[\ell\Big(\boldsymbol{\theta}\Big) + \log p(\boldsymbol{\theta}) + C\Big], \qquad (C \text{ is independent of } \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \Big[\ell\Big(\boldsymbol{\theta}\Big) + \log p(\boldsymbol{\theta})\Big]. \end{aligned} \tag{46}$$

In the case of a uniform (constant and thus improper) prior distribution, $p(\boldsymbol{\theta}) =$

C, the MAP estimate reduces to the *maximum likelihood* (ML) estimate

$$\boldsymbol{\theta}_{\mathrm{ML}} \equiv \arg\max_{\boldsymbol{\theta}} \Big[\ell\Big(\boldsymbol{\theta}\Big)\Big]. \tag{47}$$

In the limit of infinite data, the influence of the prior disappears. Then if the support of the prior includes the true parameter value, the MAP estimate has the same asymptotic properties as the ML estimate (Cappé et al., 2005). Since the mathematical difference between the MAP and ML estimates depends only on the model dependent prior distribution assigned to $\boldsymbol{\theta}$, we will mainly focus on computing the ML estimate. Some steps where the prior plays an important role will be separately highlighted.

With the help of the Gaussian filtering and smoothing methodology introduced in Section 3.2, computing the (approximate) MAP estimate corresponds to maximizing a completely known function. Thus the problem is turned into one of nonlinear optimization (also called nonlinear programming) (Cappé et al., 2005).

4.1.2 Ascent methods

Both of the parameter estimation methods we are going discuss, the expectation maximization algorithm and the instances of gradient based nonlinear programming dealt with in the next chapter, belong to the class of *iterative ascent methods* (Luenberger & Ye, 2008). Suppose that $\mathbf{m} : \Theta \to \Theta$ defines an iterative ascent method and that we are maximizing the objective function $\ell : \Theta \to \mathbb{R}$. Then given some initial point $\boldsymbol{\theta}_0$, the sequence of estimates $\{\boldsymbol{\theta}_j \in \Theta : \boldsymbol{\theta}_j = \mathbf{m}(\boldsymbol{\theta}_{j-1})\}$ where $j = 1, \ldots$ has the property $\ell\Big(\boldsymbol{\theta}_j\Big) \geq \ell\Big(\boldsymbol{\theta}_{j-1}\Big)$. This means that the objective function is increased at every iteration of an iterative ascent method. Given some regularity and boundedness conditions, it also means that objective function necessarily converges to a local maximum (Cappé et al., 2005; Luenberger & Ye, 2008).

4.2 Gradient based nonlinear optimization

There exists a large amount of efficient nonlinear optimization methods that require the gradient of the objective function to be available (Luenberger & Ye, 2008). The best known general purpose algorithms probably belong to the classes of quasi-Newton or conjugate gradient methods. For example, the MATLAB Optimization Toolbox contains the function `fminunc` utilizing both conjugate gradient and quasi-Newton methods in certain cases (The Mathworks Inc. 2012).

The simplest gradient based method is the *method of steepest ascent*. It requires that the first partial derivatives of the objective function are defined and continuous in their domain. The method of steepest ascent is then defined by the iteration

$$\boldsymbol{\theta}_{j+1} = \boldsymbol{\theta}_j + \alpha_j \nabla \ell\Big(\boldsymbol{\theta}_j\Big). \tag{48}$$

The idea is intuitive since it is well known that the gradient points to the direction of steepest ascent, a direction that is orthogonal to the isolines of constant value. To determine α_j, the *step size*, another minimization problem needs to be solved, namely

$$\alpha_j = \arg\min_{\alpha} \ell\Big(\boldsymbol{\theta}_j + \alpha \nabla \ell\Big(\boldsymbol{\theta}_j\Big)\Big). \tag{49}$$

The one dimensional optimization algorithms that are used to solve the step-sizes are known as *line search* methods (Luenberger & Ye, 2008). Common line search methods include the golden rule method and methods based on polynomial interpolation.

Suppose now that $\boldsymbol{\theta}_\star$ is the value of the parameter giving the unique maximum of $\ell\Big(\boldsymbol{\theta}\Big)$. We define the *order of convergence* as the supremum of the numbers $p \geq 0$, where

$$0 \geq \lim_{j\to\infty} \frac{\Big|\boldsymbol{\theta}_{j+1} - \boldsymbol{\theta}_\star\Big|}{\Big|\boldsymbol{\theta}_j - \boldsymbol{\theta}_\star\Big|^p} < \infty. \tag{50}$$

When $p = 1$, we also define the *linear rate of convergence* as the number $0 \leq \rho < 1$ in

$$\lim_{j\to\infty} \frac{\Big|\boldsymbol{\theta}_{j+1} - \boldsymbol{\theta}_\star\Big|}{\Big|\boldsymbol{\theta}_j - \boldsymbol{\theta}_\star\Big|} = \rho. \tag{51}$$

It can be shown that the steepest ascent method has linear order of convergence ($p = 1$) and if the Hessian of the objective function is positive definite with $r = A/a$, the ratio of the largest and smallest eigenvalues,

$$\rho \leq \left(\frac{r-1}{r+1}\right)^2. \tag{52}$$

A much more efficient nonlinear optimization algorithm is the *Newton's method*. It is based on Taylor expanding the objective function around the current estimate $\boldsymbol{\theta}_j$.

Let us assume that ℓ has continuous second-order partial derivatives. Then

$$\ell\Big(\boldsymbol{\theta}\Big) \approx \ell\Big(\boldsymbol{\theta}_j\Big) + \nabla\ell\Big(\boldsymbol{\theta}_j\Big)^{\mathsf{T}}\Big(\boldsymbol{\theta} - \boldsymbol{\theta}_j\Big) + \frac{1}{2}\Big(\boldsymbol{\theta} - \boldsymbol{\theta}_j\Big)^{\mathsf{T}}\nabla^2\ell\Big(\boldsymbol{\theta}_j\Big)\Big(\boldsymbol{\theta} - \boldsymbol{\theta}_j\Big)$$

and maximizing the approximation by setting its gradient to zero gives

$$\begin{aligned}\nabla\ell\Big(\boldsymbol{\theta}_j\Big) - \nabla^2\ell\Big(\boldsymbol{\theta}_j\Big)\Big(\boldsymbol{\theta}_j - \boldsymbol{\theta}\Big) = 0 \\ \Rightarrow \boldsymbol{\theta}_{j+1} = \boldsymbol{\theta}_j - \nabla^2\ell\Big(\boldsymbol{\theta}_j\Big)^{-1}\nabla\ell\Big(\boldsymbol{\theta}_j\Big).\end{aligned} \tag{53}$$

Near $\boldsymbol{\theta}_\star$ the Hessian is invertible and so the algorithm is well defined there (see,e.g., Luenberger & Ye, 2008). It can be shown that when initialized sufficiently close to $\boldsymbol{\theta}_\star$, (pure form) Newton's method always converges to $\boldsymbol{\theta}_\star$ with order of convergence at least *two*.

Further away from the maximum, there are various problems with Newton's method as formulated in Equation (53). There are no guarantees for the invertibility of the Hessian and higher order terms may cause a step to actually decrease the objective function. Thus we turn our attention to algorithms of the general form

$$\boldsymbol{\theta}_{j+1} = \boldsymbol{\theta}_j - \widehat{\mathbf{H}}_j^{-1}\nabla\ell\Big(\boldsymbol{\theta}_j\Big), \tag{54}$$

where $\widehat{\mathbf{H}}_j$ is a symmetric matrix, the *search direction* is $\mathbf{D}_j\nabla\ell\Big(\boldsymbol{\theta}_j\Big)$ and the *step-size* is $\alpha_j > 0$. Generally $\widehat{\mathbf{H}}_j$ should also be negative definite, to guarantee that the method is an ascent method for small α_j.

Clearly we get gradient ascent with $\widehat{\mathbf{H}}_j = \mathbf{I}$ and Newton's method with $\widehat{\mathbf{H}}_j = \nabla^2\ell\Big(\boldsymbol{\theta}_j\Big)$. Other methods of this form have thus orders of convergence between one and two. In practice the step size parameter is always determined by a line-search, so that different algorithms of the form (54) differ only in how the search direction is computed. Even if we could guarantee the invertibility of the Hessian, its computation is nevertheless notoriously computationally demanding.

Thus we will discuss methods derived from Newton's method, but which only require gradient information. These are commonly known as *quasi-Newton* methods or sometimes *secant methods* (Battiti, 1992). Given the analytical gradient, the idea is to *iteratively* approximate the analytical inverse Hessian by utilizing information gathered as the ascent method advances. Suppose we are given two points, $\boldsymbol{\theta}_j$ and

$\boldsymbol{\theta}_{j+1}$, and that

$$\mathbf{g}_j \equiv \nabla\ell\Big(\boldsymbol{\theta}_j\Big) \tag{55}$$

$$\mathbf{q}_j \equiv \mathbf{g}_{j+1} - \mathbf{g}_j \tag{56}$$

$$\mathbf{p}_j \equiv \boldsymbol{\theta}_{j+1} - \boldsymbol{\theta}_j. \tag{57}$$

We could then approximate the Hessian from

$$\mathbf{q}_j \approx \nabla^2\ell\Big(\boldsymbol{\theta}_j\Big)\mathbf{p}_j, \tag{58}$$

which in the one dimensional case is the slope of the secant line drawn through the two points θ_j and θ_{j+1} (Battiti, 1992). In case of constant Hessian, Equation (58) becomes exact. In multiple dimensions Equation 58 doesn't give a unique solution for the approximate Hessian. The Broyden *update* suggests to pick the one that deviates the least from the current approximation in the sense of the Frobenius norm. Let us suppose that we're searching for a symmetric and negative definite approximate Hessian $\widehat{\mathbf{H}}_{j+1}$ based on the current approximation $\widehat{\mathbf{H}}_j$. Since the Broyden update doesn't guarantee negative definiteness we instead update an *invertible* Cholesky factor, thus guaranteeing the negative-definiteness of $\widehat{\mathbf{H}}_{j+1}$. These considerations lead to the widely applied *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) (Broyden, Dennis, & Moré, 1973; Battiti, 1992) update

$$\widehat{\mathbf{H}}_{j+1} = \widehat{\mathbf{H}}_j + \frac{\mathbf{q}_k\mathbf{q}_k^\mathsf{T}}{\mathbf{q}_k^\mathsf{T}\mathbf{p}_k} + \frac{\widehat{\mathbf{H}}_j\mathbf{p}_k\mathbf{p}_k^\mathsf{T}\widehat{\mathbf{H}}_j}{\mathbf{p}_k^\mathsf{T}\widehat{\mathbf{H}}_j\mathbf{p}_k}. \tag{59}$$

Since we are actually in need for the approximate *inverse* Hessian, applying the Sherman-Morrison inversion formula gives

$$\widehat{\mathbf{H}}_{j+1}^{-1} = \widehat{\mathbf{H}}_j^{-1} + \left(\frac{1 + \mathbf{q}_k^\mathsf{T}\widehat{\mathbf{H}}_j^{-1}\mathbf{q}_k}{\mathbf{q}_k^\mathsf{T}\mathbf{q}_k}\right)\frac{\mathbf{p}_k\mathbf{p}_k^\mathsf{T}}{\mathbf{p}_k^\mathsf{T}\mathbf{q}_k} + \frac{\mathbf{p}_k\mathbf{q}_k^\mathsf{T}\widehat{\mathbf{H}}_j^{-1} + \widehat{\mathbf{H}}_j^{-1}\mathbf{q}_k\mathbf{q}_k^\mathsf{T}}{\mathbf{q}_k^\mathsf{T}\mathbf{p}_k}. \tag{60}$$

It should be pointed that the commonly used MATLAB unconditional nonlinear optimization function `fminunc` that we referred to earlier, uses the BFGS quasi-Newton method with cubic (and occasionally quadratic) polynomial interpolation based line search.

4.2.1 Linear-Gaussian SSMs

Let us then focus on computing the gradient of the log-likelihood function $\ell\big(\boldsymbol{\theta}\big)$, also known as the *score function*. By marginalizing the joint distribution of Equation (22) we get

$$p(\mathbf{y}_k \,|\, \mathbf{y}_{1:k-1}, \boldsymbol{\theta}) = \mathrm{N}\Big(\mathbf{y}_k \,\Big|\, \mathbf{H}\mathbf{m}_{k|k-1}, \mathbf{S}_k\Big). \tag{61}$$

Applying Equation (17) and taking the logarithm then gives

$$\ell\big(\boldsymbol{\theta}\big) = -\frac{1}{2}\sum_{k=1}^{T} \log\det \mathbf{S}_k - \frac{1}{2}\sum_{k=1}^{T} \Big(\mathbf{y}_k - \mathbf{H}\mathbf{m}_{k|k-1}\Big)^{\mathsf{T}} \mathbf{S}_k^{-1} \Big(\mathbf{y}_k - \mathbf{H}\mathbf{m}_{k|k-1}\Big) + C, \tag{62}$$

where C is a constant that doesn't depend on $\boldsymbol{\theta}$ and thus can be ignored in the maximization. There are two seemingly quite different methods for computing the score function. The first one proceeds straightforwardly by taking the partial derivatives of $\ell\big(\boldsymbol{\theta}\big)$. As will soon be demonstrated, this leads to some additional recursive formulas, known as the *sensitivity equations*, which allow computing the gradient in parallel with the Kalman filter. The second method needs the smoothing distributions with the cross-timestep covariances and it can be easily computed with the expectation maximization machinery that will be introduced later. When applied to linear-Gaussian SSMs these two methods can be proved to compute the exact same quantity (Cappé et al., 2005). At this point we will focus on the sensitivity equations. Going further it will be assumed that $\ell\big(\boldsymbol{\theta}\big)$ is continuous and differentiable for all $\boldsymbol{\theta} \in \Theta$. We will also assume here that $\mathbf{H}$ is *independent* of $\boldsymbol{\theta}$, since in practice this is often the case (i.e., the linear mapping from the state to the measurement is known).

In order to calculate the score function

$$\nabla\ell\big(\boldsymbol{\theta}'\big) = \left.\frac{\partial\ell\big(\boldsymbol{\theta}\big)}{\partial\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}'} = \left.\left[\frac{\partial\ell\big(\boldsymbol{\theta}\big)}{\partial\theta_1} \quad \cdots \quad \frac{\partial\ell\big(\boldsymbol{\theta}\big)}{\partial\theta_{d_\theta}}\right]^{\mathsf{T}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}'}, \tag{63}$$

we have to compute the partial derivatives:

$$\begin{aligned}\frac{\partial \ell\left(\boldsymbol{\theta}\right)}{\partial \theta_i} =& -\frac{1}{2}\sum_{k=1}^{T}\operatorname{Tr}\left(\mathbf{S}_k^{-1}\frac{\partial \mathbf{S}_k}{\partial \theta_i}\right) \\ &+\sum_{k=1}^{T}\left(\mathbf{H}\frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i}\right)^{\mathsf{T}}\mathbf{S}_k^{-1}\left(\mathbf{y}_k-\mathbf{H}\mathbf{m}_{k|k-1}\right) \\ &+\frac{1}{2}\sum_{k=1}^{T}\left(\mathbf{y}_k-\mathbf{H}\mathbf{m}_{k|k-1}\right)^{\mathsf{T}}\mathbf{S}_k^{-1}\left(\frac{\partial \mathbf{S}_k}{\partial \theta_i}\right)\mathbf{S}_k^{-1}\left(\mathbf{y}_k-\mathbf{H}\mathbf{m}_{k|k-1}\right).\end{aligned} \tag{64}$$

From the Kalman filter recursions (21) we get

$$\frac{\partial \mathbf{S}_k}{\partial \theta_i} = \mathbf{H}\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i}\mathbf{H}^{\mathsf{T}} + \frac{\partial \mathbf{R}}{\partial \theta_i}, \tag{65}$$

so that we are left with the task of determining the partial derivatives of $\mathbf{m}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$,

$$\frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i} = \frac{\partial \mathbf{A}}{\partial \theta_i}\mathbf{m}_{k-1|k-1} + \mathbf{A}\frac{\partial \mathbf{m}_{k-1|k-1}}{\partial \theta_i} \tag{66}$$

$$\begin{aligned}\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} =& \frac{\partial \mathbf{A}}{\partial \theta_i}\mathbf{P}_{k-1|k-1}\mathbf{A}^{\mathsf{T}} + \mathbf{A}\frac{\partial \mathbf{P}_{k-1|k-1}}{\partial \theta_i}\mathbf{A}^{\mathsf{T}} \\ &+\mathbf{A}\mathbf{P}_{k-1|k-1}\left(\frac{\partial \mathbf{A}}{\partial \theta_i}\right)^{\mathsf{T}} + \frac{\partial \mathbf{Q}}{\partial \theta_i},\end{aligned} \tag{67}$$

as well as of $\mathbf{m}_{k|k}$ and $\mathbf{P}_{k|k}$:

$$\frac{\partial \mathbf{K}_k}{\partial \theta_i} = \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i}\mathbf{H}^{\mathsf{T}}\mathbf{S}_k^{-1} - \mathbf{P}_{k|k-1}\mathbf{H}^{\mathsf{T}}\mathbf{S}_k^{-1}\frac{\partial \mathbf{S}_k}{\partial \theta_i}\mathbf{S}_k^{-1} \tag{68}$$

$$\frac{\partial \mathbf{m}_{k|k}}{\partial \theta_i} = \frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{K}_k}{\partial \theta_i}\left(\mathbf{y}_k-\mathbf{H}\mathbf{m}_{k|k-1}\right) - \mathbf{K}_k\mathbf{H}\frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i} \tag{69}$$

$$\frac{\partial \mathbf{P}_{k|k}}{\partial \theta_i} = \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} - \frac{\partial \mathbf{K}_k}{\partial \theta_i}\mathbf{S}_k\mathbf{K}_k^{\mathsf{T}} - \mathbf{K}_k\frac{\partial \mathbf{S}_k}{\partial \theta_i}\mathbf{K}_k^{\mathsf{T}} - \mathbf{K}_k\mathbf{S}_k\left(\frac{\partial \mathbf{K}_k}{\partial \theta_i}\right)^{\mathsf{T}}. \tag{70}$$

Equations (66), (67), (68), (69) and (70) together specify a recursive algorithm for computing (64) that can be run alongside the Kalman filter recursions. As noted earlier, these equations are sometimes known as the *sensitivity equations* and they are derived at least in Gupta and Mehra (1974). See also Sandell and Yared (1978) and Mbalawata, Särkkä, and Haario (2012).

4.2.2 Nonlinear-Gaussian SSMs

Here we will present the derivation of the sensitivity equations for nonlinear SSMs with additive Gaussian noise. Since the predictive and filtering distributions have to be approximated in the nonlinear case, we will work in the Gaussian filtering framework. The 3rd order spherical cubature approximation of Equation (44) will be applied to integrals intractable in closed form. The result is an approximate recursive algorithm for computing $\frac{\partial \mathbf{m}_{k|k}}{\partial \theta_i}$ and $\frac{\partial \mathbf{P}_{k|k}}{\partial \theta_i}$, which are the partial derivatives of the mean and and variance of the filtering distributions. These enable us to compute the partial derivatives of the marginal log-likelihood and by Equation (63), an approximation to the score function.

By marginalizing the joint distribution of Equation (33) we get the approximation

$$p(\mathbf{y}_k \,|\, \mathbf{y}_{1:k-1}, \boldsymbol{\theta}) \approx \mathrm{N}(\mathbf{y}_k \,|\, \boldsymbol{\mu}_k, \mathbf{S}_k), \tag{71}$$

so that taking the logarithm of the factorization (17) gives the approximate log marginal likelihood

$$\ell\Big(\boldsymbol{\theta}\Big) \approx -\frac{1}{2}\sum_{k=1}^{T} \log\det \mathbf{S}_k - \frac{1}{2}\sum_{k=1}^{T} \left(\mathbf{y}_k - \boldsymbol{\mu}_k\right)^{\mathsf{T}} \mathbf{S}_k^{-1} \left(\mathbf{y}_k - \boldsymbol{\mu}_k\right), \tag{72}$$

where terms independent of $\boldsymbol{\theta}$ have been dropped. To compute the score function, we need the partial derivatives

$$\begin{aligned} \frac{\partial \ell\Big(\boldsymbol{\theta}\Big)}{\partial \theta_i} \approx & -\frac{1}{2}\sum_{k=1}^{T} \mathrm{Tr}\left(\mathbf{S}_k^{-1}\frac{\partial \mathbf{S}_k}{\partial \theta_i}\right) \\ & + \sum_{k=1}^{T} \left(\frac{\partial \boldsymbol{\mu}_k}{\partial \theta_i}\right)^{\mathsf{T}} \mathbf{S}_k^{-1} \left(\mathbf{y}_k - \boldsymbol{\mu}_k\right) \\ & + \frac{1}{2}\sum_{k=1}^{T} \left(\mathbf{y}_k - \boldsymbol{\mu}_k\right)^{\mathsf{T}} \mathbf{S}_k^{-1} \left(\frac{\partial \mathbf{S}_k}{\partial \theta_i}\right) \mathbf{S}_k^{-1} \left(\mathbf{y}_k - \boldsymbol{\mu}_k\right). \end{aligned} \tag{73}$$

Let us denote the predictive distribution sigma points by $\boldsymbol{\varsigma}_{k|k-1}^{(j)} = \mathbf{m}_{k|k-1} + \sqrt{\mathbf{P}_{k|k-1}}\boldsymbol{\varepsilon}^{(j)}$, where $j = 1, \ldots, 2\,d_x$, and the constant weight by $w = \frac{1}{2d_x}$. We will first focus on computing an approximation to

$$\frac{\partial \boldsymbol{\varsigma}_{k|k-1}^{(j)}}{\partial \theta_i} = \frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i} + \frac{\partial \sqrt{\mathbf{P}_{k|k-1}}}{\partial \theta_i}\,\boldsymbol{\varepsilon}^{(j)}. \tag{74}$$

By applying the cubature rule to the integrals (30) and (31) we get

$$\mathbf{m}_{k|k-1} \approx w \sum_{j=1}^{2\,d_x} \mathbf{f}\left(\boldsymbol{\varsigma}_{k-1|k-1}^{(j)}\right) \tag{75}$$

$$\mathbf{P}_{k|k-1} \approx w \sum_{j=1}^{2\,d_x} \left(\mathbf{f}\left(\boldsymbol{\varsigma}_{k-1|k-1}^{(j)}\right) - \mathbf{m}_{k|k-1}\right)\left(\mathbf{f}\left(\boldsymbol{\varsigma}_{k-1|k-1}^{(j)}\right) - \mathbf{m}_{k|k-1}\right)^{\mathsf{T}} + \mathbf{Q}, \tag{76}$$

so that the partial derivatives of these become

$$\frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i} \approx w \sum_{j=1}^{2\,d_x} \mathbf{J}_{\mathbf{f}}\left(\boldsymbol{\varsigma}_{k-1|k-1}^{(j)}\right) \frac{\partial \boldsymbol{\varsigma}_{k-1|k-1}^{(j)}}{\partial \theta_i} \tag{77}$$

and

$$\begin{aligned}
&\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} \approx \\
&\quad w \sum_{j=1}^{2\,d_x} \Bigg[\left(\mathbf{J}_{\mathbf{f}}\left(\boldsymbol{\varsigma}_{k-1|k-1}^{(j)}\right) \frac{\partial \boldsymbol{\varsigma}_{k-1|k-1}^{(j)}}{\partial \theta_i} - \frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i}\right)\left(\mathbf{f}\left(\boldsymbol{\varsigma}_{k-1|k-1}^{(j)}\right) - \mathbf{m}_{k|k-1}\right)^{\mathsf{T}} \\
&\qquad + \left(\mathbf{f}\left(\boldsymbol{\varsigma}_{k-1|k-1}^{(j)}\right) - \mathbf{m}_{k|k-1}\right)\left(\mathbf{J}_{\mathbf{f}}\left(\boldsymbol{\varsigma}_{k-1|k-1}^{(j)}\right) \frac{\partial \boldsymbol{\varsigma}_{k-1|k-1}^{(j)}}{\partial \theta_i} - \frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i}\right)^{\mathsf{T}} \Bigg] \\
&\qquad + \frac{\partial \mathbf{Q}}{\partial \theta_i}\,.
\end{aligned} \tag{78}$$

Here $\mathbf{J}_{\mathbf{f}}(\cdot)$ denotes the Jacobian of $\mathbf{f}$. We assume that at the current iteration k we have available the approximate mean and variance of the previous filtering distribution, $\mathbf{m}_{k-1|k-1}$ and $\mathbf{P}_{k-1|k-1}$, as well as the partial derivatives $\frac{\partial \mathbf{m}_{k-1|k-1}}{\partial \theta_i}$ and $\frac{\partial \mathbf{P}_{k-1|k-1}}{\partial \theta_i}$. This means we can form $\boldsymbol{\varsigma}_{k-1|k-1}^{(j)} = \mathbf{m}_{k-1|k-1} + \sqrt{\mathbf{P}_{k-1|k-1}}\boldsymbol{\varepsilon}^{(j)}$.

In Equation (74) we clearly need $\frac{\partial \sqrt{\mathbf{P}_{k|k-1}}}{\partial \theta_i}$, the partial derivative of the Cholesky decomposition of $\mathbf{P}_{k|k-1}$. Having $\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i}$ available, this can be obtained for example by differentiating an algorithm for computing the Cholesky decomposition.

By applying the CKF cubature rule to the integral (35) we get

$$\mathbf{S}_k \approx w \sum_{j=1}^{2\,d_x} \left(\mathbf{h}\left(\boldsymbol{\varsigma}_{k|k-1}^{(j)}\right) - \boldsymbol{\mu}_k\right)\left(\mathbf{h}\left(\boldsymbol{\varsigma}_{k|k-1}^{(j)}\right) - \boldsymbol{\mu}_k\right)^{\mathsf{T}} + \mathbf{R}, \tag{79}$$

so that

$$\frac{\partial \mathbf{S}_k}{\partial \theta_i} \approx w \sum_{j=1}^{2\,d_x} \Bigg[\Big(\mathbf{J}_{\mathbf{h}}\big(\boldsymbol{\varsigma}_{k|k-1}^{(j)}\big) \frac{\partial \boldsymbol{\varsigma}_{k|k-1}^{(j)}}{\partial \theta_i} - \frac{\partial \boldsymbol{\mu}_k}{\partial \theta_i} \Big) \Big(\mathbf{h}\big(\boldsymbol{\varsigma}_{k|k-1}^{(j)}\big) - \boldsymbol{\mu}_k \Big)^{\mathsf{T}} + \Big(\mathbf{h}\big(\boldsymbol{\varsigma}_{k|k-1}^{(j)}\big) - \boldsymbol{\mu}_k \Big) \Big(\mathbf{J}_{\mathbf{h}}\big(\boldsymbol{\varsigma}_{k|k-1}^{(j)}\big) \frac{\partial \boldsymbol{\varsigma}_{k|k-1}^{(j)}}{\partial \theta_i} - \frac{\partial \boldsymbol{\mu}_k}{\partial \theta_i} \Big)^{\mathsf{T}} \Bigg] + \frac{\partial \mathbf{R}}{\partial \theta_i}, \tag{80}$$

where $\mathbf{J}_{\mathbf{h}}(\cdot)$ denotes the Jacobian of $\mathbf{h}$. The approximate partial derivative of $\boldsymbol{\mu}_k$ can be derived from Equation (34):

$$\frac{\partial \boldsymbol{\mu}_k}{\partial \theta_i} \approx w \sum_{j=1}^{2\,d_x} \mathbf{J}_{\mathbf{h}}\big(\boldsymbol{\varsigma}_{k|k-1}^{(j)}\big) \frac{\partial \boldsymbol{\varsigma}_{k|k-1}^{(j)}}{\partial \theta_i}. \tag{81}$$

From Equation (37b) we get

$$\frac{\partial \mathbf{K}_k}{\partial \theta_i} = \frac{\partial \mathbf{C}_k}{\partial \theta_i} \mathbf{S}_k^{-1} - \mathbf{C}_k \mathbf{S}_k^{-1} \frac{\partial \mathbf{S}_k}{\partial \theta_i} \mathbf{S}_k^{-1} \tag{82}$$

and Equation (36) gives

$$\mathbf{C}_k \approx w \sum_{j=1}^{2\,d_x} \Big(\boldsymbol{\varsigma}_{k|k-1}^{(j)} - \mathbf{m}_{k|k-1} \Big) \Big(\mathbf{h}\big(\boldsymbol{\varsigma}_{k|k-1}^{(j)}\big) - \boldsymbol{\mu}_k \Big)^{\mathsf{T}}, \tag{83}$$

so that

$$\frac{\partial \mathbf{C}_k}{\partial \theta_i} \approx w \sum_{j=1}^{2\,d_x} \Bigg[\Big(\frac{\partial \boldsymbol{\varsigma}_{k|k-1}^{(j)}}{\partial \theta_i} - \frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i} \Big) \Big(\mathbf{h}\big(\boldsymbol{\varsigma}_{k|k-1}^{(j)}\big) - \boldsymbol{\mu}_k \Big)^{\mathsf{T}} + \Big(\boldsymbol{\varsigma}_{k|k-1}^{(j)} - \mathbf{m}_{k|k-1} \Big) \Big(\mathbf{J}_{\mathbf{h}}\big(\boldsymbol{\varsigma}_{k|k-1}^{(j)}\big) \frac{\partial \boldsymbol{\varsigma}_{k|k-1}^{(j)}}{\partial \theta_i} - \frac{\partial \boldsymbol{\mu}_k}{\partial \theta_i} \Big)^{\mathsf{T}} \Bigg]. \tag{84}$$

Finally, from Equations (37c) and (37d) we obtain

$$\frac{\partial \mathbf{m}_{k|k}}{\partial \theta_i} = \frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{K}_k}{\partial \theta_i} (\mathbf{y}_k - \boldsymbol{\mu}_k) - \mathbf{K}_k \frac{\partial \boldsymbol{\mu}_k}{\partial \theta_i}. \tag{85}$$

and

$$\frac{\partial \mathbf{P}_{k|k}}{\partial \theta_i} = \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} - \frac{\partial \mathbf{K}_k}{\partial \theta_i} \mathbf{S}_k \mathbf{K}_k^{\mathsf{T}} - \mathbf{K}_k \frac{\partial \mathbf{S}_k}{\partial \theta_i} \mathbf{K}_k^{\mathsf{T}} - \mathbf{K}_k \mathbf{S}_k \left(\frac{\partial \mathbf{K}_k}{\partial \theta_i} \right)^{\mathsf{T}} \tag{86}$$

4.3 Expectation maximization (EM)

Expectation maximization (EM) algorithm is a general method for finding ML and MAP estimates in probabilistic models with missing data or latent variables. It was first introduced in the celebrated article of Dempster and Laird (1977) and its convergence properties were proved in C. F. J. Wu (1983) . Instead of maximizing (62) directly, EM alternates between computing a variational lower bound and then maximizing this bound (Bishop, 2006; Barber, 2012). As will be seen, since the bound is strict, increasing the bound implies an increase in the objective function. We shall use $\langle\cdot\rangle_q \equiv \int \cdot\, q(z)\,\mathrm{d}z$ to denote the expectation over any distribution $q(z)$.

Let us introduce a family of "variational" distributions indexed by the parameter $\boldsymbol{\psi}$, $q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi})$, over the states $\mathbf{x}_{0:T}$ (or the latent variables in general). Noting now that $p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}) = p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T} \,|\, \boldsymbol{\theta})/p(\mathbf{y}_{0:T} \,|\, \boldsymbol{\theta})$ and that $\ell\Big(\boldsymbol{\theta}\Big) \equiv \log p(\mathbf{y}_{0:T} \,|\, \boldsymbol{\theta})$ is independent of $\mathbf{x}_{0:T}$, we can perform the following decomposition on the marginal log likelihood:

$$\begin{aligned}
\ell\Big(\boldsymbol{\theta}\Big) &= \log p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T} \,|\, \boldsymbol{\theta}) - \log p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}) \\
&= \big\langle \log p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T} \,|\, \boldsymbol{\theta}) \big\rangle_{q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi})} - \big\langle \log p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}) \big\rangle_{q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi})} \\
&= \underbrace{\big\langle \log p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T} \,|\, \boldsymbol{\theta}) \big\rangle_{q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi})} - \big\langle q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi}) \big\rangle_{q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi})}}_{\mathcal{B}\Big(\boldsymbol{\theta}, \boldsymbol{\psi}\Big)} \\
&\quad + \mathrm{KL}\Big(q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi}) \,\Big\|\, p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta})\Big).
\end{aligned} \tag{87}$$

By invoking the nonnegativeness of the *Kullback-Leibler divergence*

$$\mathrm{KL}\Big(q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi}) \,\Big\|\, p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta})\Big) = -\left\langle \log \frac{p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta})}{q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi})} \right\rangle_{q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi})}, \tag{88}$$

or equivalently the relation

$$\big\langle \log q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi}) \big\rangle_{q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi})} \geq \big\langle \log p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}) \big\rangle_{q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi})}, \tag{89}$$

provable by *Jensen's inequality*, we see that $\mathcal{B}\Big(\boldsymbol{\theta}, \boldsymbol{\psi}\Big)$ is indeed a lower bound on $\ell\Big(\boldsymbol{\theta}\Big)$. These considerations suggest an iterative algorithm which produces a series of estimates $\{\boldsymbol{\theta}_j\}$, where $j = 0, \ldots$. Given the initial guess $\boldsymbol{\theta}_0$, the two alternating steps of the algorithm are:

E-step

Set $q\Big(\mathbf{x}_{0:T} \,\Big|\, \boldsymbol{\psi}_{j+1}\Big)$ to the distribution that maximizes $\mathcal{B}\Big(\boldsymbol{\theta}_j, \boldsymbol{\psi}\Big)$ with re-

spect to $\boldsymbol{\psi}$. Here $\boldsymbol{\theta}_j$ is the current estimate of $\boldsymbol{\theta}$.

M-step

Set $\boldsymbol{\theta}_{j+1}$ to the estimate that maximizes $\mathcal{B}\big(\boldsymbol{\theta}, \boldsymbol{\psi}_{j+1}\big)$ with respect to $\boldsymbol{\theta}$.

In some sense then, the algorithm can be viewed as coordinate ascent in $\mathcal{B}\big(\boldsymbol{\theta}, \boldsymbol{\psi}\big)$ (Neal & Hinton, 1998).

The *sharpest* bound can clearly be found among distributions of the form $p\Big(\mathbf{x}_{0:T} \,\Big|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}'\Big)$, since the Kullback-Leibler divergence vanishes with $q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi}) = p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta})$. Let us now define

$$\mathcal{Q}\big(\boldsymbol{\theta}, \boldsymbol{\theta}'\big) \equiv \langle \log p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T} \,|\, \boldsymbol{\theta}) \rangle_{p\left(\mathbf{x}_{0:T} \,\middle|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}'\right)} \tag{90}$$

$$\mathcal{H}\big(\boldsymbol{\theta}, \boldsymbol{\theta}'\big) \equiv \langle \log p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}) \rangle_{p\left(\mathbf{x}_{0:T} \,\middle|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}'\right)} \tag{91}$$

$$\mathcal{B}\big(\boldsymbol{\theta}, \boldsymbol{\theta}'\big) \equiv \mathcal{Q}\big(\boldsymbol{\theta}, \boldsymbol{\theta}'\big) - \mathcal{H}\big(\boldsymbol{\theta}', \boldsymbol{\theta}'\big). \tag{92}$$

Regarding these functions we will use the convention that denoting for example $\mathcal{B}\big(\boldsymbol{\theta}, \boldsymbol{\psi}\big)$ means the expectation is taken with respect to some *unspecified* distribution $q(\mathbf{x}_{0:T} \,|\, \boldsymbol{\psi})$, whereas $\mathcal{B}\big(\boldsymbol{\theta}, \boldsymbol{\theta}'\big)$ implies it is taken with respect to $p\Big(\mathbf{x}_{0:T} \,\Big|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}'\Big)$, meaning the posterior distribution of the states given the parameter $\boldsymbol{\theta}'$.

According to (87) we now have

$$\ell\big(\boldsymbol{\theta}\big) \geq \mathcal{B}\big(\boldsymbol{\theta}, \boldsymbol{\theta}'\big) \quad \forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta \tag{93}$$

and especially

$$\ell\big(\boldsymbol{\theta}\big) = \mathcal{B}\big(\boldsymbol{\theta}, \boldsymbol{\theta}\big). \tag{94}$$

When we want to maximize $\mathcal{B}\big(\boldsymbol{\theta}, \boldsymbol{\theta}'\big)$ with respect to $\boldsymbol{\theta}$, it clearly suffices to consider only $\mathcal{Q}\big(\boldsymbol{\theta}, \boldsymbol{\theta}'\big)$, known as the *expected complete-data log-likelihood* or the *intermediate quantity of EM* (Cappé et al., 2005; Bishop, 2006).

What is also interesting about $\mathcal{Q}\big(\boldsymbol{\theta}, \boldsymbol{\theta}'\big)$ is that it can be used to compute the gradient of the log-likelihood, meaning the score, itself. From Equations (92) and (94) it can be seen rather easily that the score evaluated at $\widehat{\boldsymbol{\theta}}$ is given by

$$\nabla \ell\big(\widehat{\boldsymbol{\theta}}\big) = \nabla_{\boldsymbol{\theta}} \mathcal{Q}\big(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\theta}}\big) \equiv \left. \frac{\partial \mathcal{Q}\big(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}\big)}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta} = \widehat{\boldsymbol{\theta}}}. \tag{95}$$

Equation (95) is known as *Fisher's identity* (Cappé et al., 2005; Segal & Wein-

stein, 1989). It gives an alternative route for the score function computation. The implications will be discussed in more detail in the sequel.

We are now in a position to formulate the so called *fundamental inequality of EM* (Cappé et al., 2005). From (93) we have

$$\ell\Big(\boldsymbol{\theta}_{j+1}\Big) \geq \mathcal{Q}\Big(\boldsymbol{\theta}_{j+1}, \boldsymbol{\theta}_j\Big) - \mathcal{H}\Big(\boldsymbol{\theta}_j, \boldsymbol{\theta}_j\Big),$$

so that using (94) and assuming that the M-step result $\boldsymbol{\theta}_{j+1}$ increases the bound we can write

$$\ell\Big(\boldsymbol{\theta}_{j+1}\Big) - \ell\Big(\boldsymbol{\theta}_j\Big) \geq \mathcal{Q}\Big(\boldsymbol{\theta}_{j+1}, \boldsymbol{\theta}_j\Big) - \mathcal{Q}\Big(\boldsymbol{\theta}_j, \boldsymbol{\theta}_j\Big) \geq 0. \tag{96}$$

This highlights the fact that *the likelihood is increased or unchanged with every new estimate* $\boldsymbol{\theta}_{j+1}$. Also following from (96) is the fact that if the iterations stop at a certain point, meaning $\boldsymbol{\theta}_l = \boldsymbol{\theta}_{l-1}$ at iteration l, then $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}_l)$ must be maximal at $\boldsymbol{\theta} = \boldsymbol{\theta}_l$ and thus its gradient, and by (95) that of the likelihood, must be zero at $\boldsymbol{\theta} = \boldsymbol{\theta}_l$. Thus $\boldsymbol{\theta}_l$ is a *stationary point* of $\ell\Big(\boldsymbol{\theta}\Big)$, that is, a local maximum or a saddle point.

Figure 3 illustrates the EM algorithm for a unidimensional parameter θ. Starting from the lower left corner, given the current parameter estimate θ_k, the E-step computes the lower bound $\mathcal{B}\Big(\theta, \theta_k\Big)$ (dashed line) to the objective function $\ell\Big(\theta\Big)$ (solid line). Clearly $\mathcal{B}\Big(\theta_k, \theta_k\Big) = \ell\Big(\theta_k\Big)$ and $\nabla_\theta \mathcal{B}\Big(\theta_k, \theta_k\Big) = \nabla_\theta \mathcal{Q}(\theta_k, \theta_k) = \nabla \ell\Big(\theta_k\Big)$. In the M-step, the next parameter value θ_{k+1} is found by maximizing the lower bound obtained in the E-step. In the case of Figure 3, we can see that EM estimate is close to the ML estimate at iteration $k + 2$.

We have so far formulated the EM algorithm only for ML estimation. In the case of MAP estimation with a nonuniform prior (remember that with a uniform prior the estimates are identical), the E-step stays the same since the prior is independent of $\mathbf{x}_{0:T}$. The MAP M-step is

M-step (MAP)

Set $\boldsymbol{\theta}_{j+1}$ to the estimate that maximizes $\mathcal{B}\Big(\boldsymbol{\theta}, \boldsymbol{\psi}_{j+1}\Big) + \log p(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

EM in exponential families of distributions

Computing the intermediate quantity of EM is especially simple if the dynamic model and the measurement model belong to an exponential family of distributions,

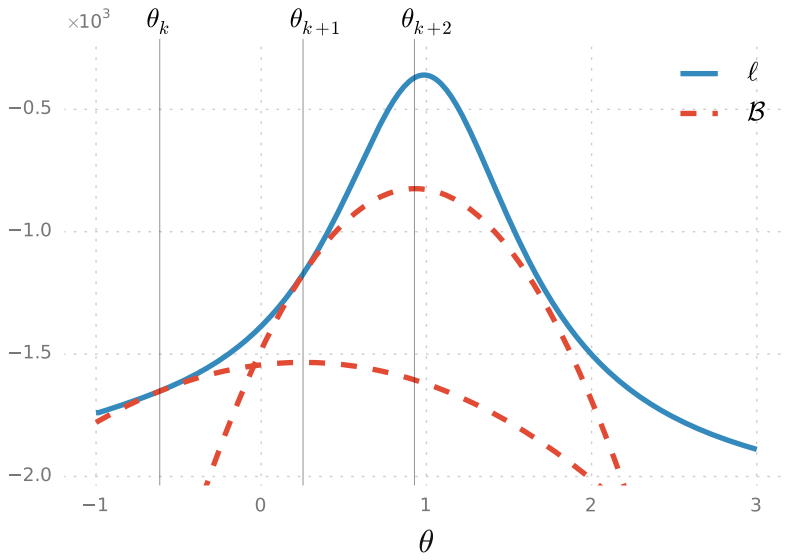

Figure 3: Illustration of two iterations of the EM algorithm for a unidimensional parameter θ. Starting from the lower left corner, given the current parameter estimate θ_k, the E-step computes the lower bound (dashed line) to the objective function $\ell\Big(\theta\Big)$ (solid line). In the M-step, the next parameter value θ_{k+1} is found by maximizing the lower bound obtained in the E-step. The EM estimate is very close to the ML estimate at iteration $k+2$.

which have probability distribution functions of the form

$$q(\mathbf{z}\,|\,\boldsymbol{\theta}) = h(\mathbf{z})\exp\Big\{\boldsymbol{\psi}(\boldsymbol{\theta})^{\mathsf{T}}\mathbf{s}(\mathbf{z}) - c(\boldsymbol{\theta})\Big\}. \tag{97}$$

Here $\mathbf{s}(\mathbf{z})$ is called the vector of *natural sufficient statistics* and $\boldsymbol{\eta} \equiv \boldsymbol{\psi}(\boldsymbol{\theta})$ is the *natural parameterization*. Let us suppose now that the complete-data likelihood is of the form (97), so that $\mathbf{z}^{\mathsf{T}} = \Big[\mathrm{vec}\{\mathbf{x}_{0:T}\}^{\mathsf{T}}, \mathrm{vec}\{\mathbf{y}_{0:T}\}^{\mathsf{T}}\Big]$, where the operator $\mathrm{vec}\{\cdot\}$ creates vectors out of matrices by stacking their columns. Thus $\mathbf{z}$ contains the hidden variables $\mathbf{x}_{0:T}$ and the measurements $\mathbf{y}_{0:T}$.

The intermediate quantity, which is the expectation of the logarithm of $q(\mathbf{z}\,|\,\boldsymbol{\theta})$ over the posterior distribution of $\mathbf{x}_{0:T}$ (implicit in the notation) becomes now

$$\mathcal{Q}\Big(\boldsymbol{\theta}, \boldsymbol{\theta}'\Big) = \boldsymbol{\psi}(\boldsymbol{\theta})^{\mathsf{T}}\langle\mathbf{s}(\mathbf{z})\rangle - c(\boldsymbol{\theta}) + \langle h(\mathbf{z})\rangle. \tag{98}$$

Since the last term is independent of $\boldsymbol{\theta}$ then the maximization in the M-step is independent of this last term. Thus the role of the E-step degenerates into computing the expectation of the sufficient statistics $\langle\mathbf{s}(\mathbf{z})\rangle$.

EM as a special case of variational Bayes

Variational Bayes (VB) is a fully Bayesian methodology where one seeks for an approximation to the parameter posterior (Barber, 2012; Bishop, 2006; MacKay, 2003; Bernardo, Bayarri, Berger, Beal, & Ghahramani, 2003)

$$p(\boldsymbol{\theta} \,|\, \mathbf{y}_{0:T}) = \frac{1}{Z} p(\mathbf{y}_{0:T} \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta}) \approx q(\boldsymbol{\theta}). \tag{99}$$

The appeal here is that when succesfull, fully Bayesian results can be obtained with significantly reduced computational requirements as compared to simulation based methods. Unfortunately it seems that applying VB to SSMs is somewhat problematic, as discussed in Turner and Sahani (2011).

Let us introduce the following simplifying factorization to the joint posterior of states and parameters:

$$p(\mathbf{x}_{0:T}, \boldsymbol{\theta} \,|\, \mathbf{y}_{0:T}) \approx q(\mathbf{x}_{0:T}) q(\boldsymbol{\theta}). \tag{100}$$

Noting now that $p(\mathbf{x}_{0:T}, \boldsymbol{\theta} \,|\, \mathbf{y}_{0:T}) = p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T}, \boldsymbol{\theta}) / p(\mathbf{y}_{0:T} \,|\, \boldsymbol{\theta})$ and that $\ell\Big(\boldsymbol{\theta}\Big) \equiv \log p(\mathbf{y}_{0:T} \,|\, \boldsymbol{\theta})$ is independent of $\mathbf{x}_{0:T}$ we can then perform the following decomposition on the log likelihood:

$$\begin{aligned}
\ell\Big(\boldsymbol{\theta}\Big) &= \log p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T}, \boldsymbol{\theta}) - \log p(\mathbf{x}_{0:T}, \boldsymbol{\theta} \,|\, \mathbf{y}_{0:T}) \\
&= \big\langle \log p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T}, \boldsymbol{\theta}) \big\rangle_{q(\mathbf{x}_{0:T}) q(\boldsymbol{\theta})} - \big\langle p(\mathbf{x}_{0:T}, \boldsymbol{\theta} \,|\, \mathbf{y}_{0:T}) \big\rangle_{q(\mathbf{x}_{0:T}) q(\boldsymbol{\theta})} \\
&= \big\langle \log p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T}, \boldsymbol{\theta}) \big\rangle_{q(\mathbf{x}_{0:T}) q(\boldsymbol{\theta})} - \big\langle q(\mathbf{x}_{0:T}) \big\rangle_{q(\mathbf{x}_{0:T})} - \big\langle q(\boldsymbol{\theta}) \big\rangle_{q(\boldsymbol{\theta})} \\
&\quad + \mathrm{KL}\Big(q(\mathbf{x}_{0:T}) q(\boldsymbol{\theta}) \,\Big\|\, p(\mathbf{x}_{0:T}, \boldsymbol{\theta} \,|\, \mathbf{y}_{0:T}) \Big).
\end{aligned} \tag{101}$$

Thus minimizing the KL divergence between the factorized approximation and the true joint posterior is equivalent to finding the tightest lower bound to the log likelihood. These considerations suggest an iterative algorithm which produces a series of estimates $q_j(\boldsymbol{\theta})$, where $j = 0, \ldots$. Given the initial guess $q_0(\boldsymbol{\theta})$, the two alternating steps of the algorithm are:

E-step

$$q_{j+1}(\mathbf{x}_{0:T}) = \operatorname*{arg\,min}_{q(\mathbf{x}_{0:T})} \mathrm{KL}\Big(q(\mathbf{x}_{0:T}) q_j(\boldsymbol{\theta}) \,\Big\|\, p(\mathbf{x}_{0:T}, \boldsymbol{\theta} \,|\, \mathbf{y}_{0:T}) \Big) \tag{102}$$

M-step

$$q_{j+1}(\boldsymbol{\theta}) = \arg\min_{q(\boldsymbol{\theta})} \mathrm{KL}\Big(q_{j+1}(\mathbf{x}_{0:T})q(\boldsymbol{\theta}) \,\Big\|\, p(\mathbf{x}_{0:T}, \boldsymbol{\theta} \,|\, \mathbf{y}_{0:T})\Big) \tag{103}$$

Let us then suppose that we only wish to find the MAP estimate $\boldsymbol{\theta}^*$. This can be accomplished by assuming a delta function form $q(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$ for the parameter factor in the joint distribution of states and parameters (100). With this assumption the bound becomes

$$p(\mathbf{y}_{0:T} \,|\, \boldsymbol{\theta}^*) \geq \langle \log p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T}, \boldsymbol{\theta}) \rangle_{q(\mathbf{x}_{0:T})q(\boldsymbol{\theta}^*)} - \langle q(\mathbf{x}_{0:T}) \rangle_{q(\mathbf{x}_{0:T})} + \texttt{const} \tag{104}$$

and the "M"-step (103) can then be written as

$$\boldsymbol{\theta}_{j+1} = \arg\max_{\boldsymbol{\theta}} \Big[\langle \log p(\mathbf{x}_{0:T}, \mathbf{y}_{0:T} \,|\, \boldsymbol{\theta}) \rangle_{q(\mathbf{x}_{0:T})} + \log p(\boldsymbol{\theta}) \Big]. \tag{105}$$

If the point estimate is plugged in the "E"-step Equation (102) we get

$$q_{j+1}(\mathbf{x}_{0:T}) \propto p\Big(\mathbf{x}_{0:T}, \mathbf{y}_{0:T} \,\Big|\, \boldsymbol{\theta}_j\Big) \propto p\Big(\mathbf{x}_{0:T} \,\Big|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}_j\Big). \tag{106}$$

Thus the EM algorithm can shown to be a special case of VB with a delta function form for $q(\boldsymbol{\theta})$.

4.3.1 Partial E and M steps

As can be seen from Equation (96), to ensure monotonicity it is enough that $\mathcal{Q}\Big(\boldsymbol{\theta}_{j+1}, \boldsymbol{\theta}_j\Big) \geq \mathcal{Q}\Big(\boldsymbol{\theta}_j, \boldsymbol{\theta}_j\Big)$, which means $\boldsymbol{\theta}_{j+1}$ is not required to be the maximum of $\mathcal{Q}\Big(\boldsymbol{\theta}, \boldsymbol{\theta}_j\Big)$. This was observed already in Dempster and Laird (1977), where methods that only seek an increase in the M-step were termed *generalized* EM (gEM) algorithms.

Another modification is the partial, or approximate, E-step. It is clear that in this case, when we cannot compute $p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{0:T}, \boldsymbol{\theta})$ exactly, the Kullback-Leibler divergence in decomposition (87) is strictly positive. This means that the lower bound we are optimizing never "touches" the log-likelihood as in Equation (94) and Figure 3. Thus EM with an approximate E step is not an ascent algorithm anymore (Goodwin & Aguero, 2005).

4.3.2 Linear-Gaussian SSMs

Let us then turn to applying EM to the case of linear-Gaussian SSMs (Shumway & Stoffer, 1982; Ghahramani, 1996) , so that

$$\begin{aligned}\mathbf{f}(\mathbf{x}_{k-1}, \boldsymbol{\theta}) &\equiv \mathbf{A}\mathbf{x}_{k-1} \\ \mathbf{h}(\mathbf{x}_k, \boldsymbol{\theta}) &\equiv \mathbf{H}\mathbf{x}_k \\ \boldsymbol{\theta} &\equiv \{\mathbf{A}, \mathbf{Q}, \mathbf{H}, \mathbf{R}\}.\end{aligned}$$

First of all, from the factorization in (3), the complete-data log-likelihood becomes

$$\begin{aligned}\ell\Big(\boldsymbol{\theta}\Big) = &-\frac{1}{2}\left(\mathbf{x}_0 - \boldsymbol{\mu}_0\right)^\mathsf{T} \boldsymbol{\Sigma}_0^{-1} \left(\mathbf{x}_0 - \boldsymbol{\mu}_0\right) - \frac{1}{2}\log\det(\boldsymbol{\Sigma}_0) \\ &- \frac{1}{2}\sum_{k=1}^{T}\left(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}\right)^\mathsf{T} \mathbf{Q}^{-1} \left(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}\right) - \frac{T}{2}\log\det(\mathbf{Q}) \\ &- \frac{1}{2}\sum_{k=1}^{T}\left(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k\right)^\mathsf{T} \mathbf{R}^{-1} \left(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k\right) - \frac{T}{2}\log\det(\mathbf{R}) \\ &+ \texttt{const}.\end{aligned} \tag{107}$$

Taking the expectation of (107) with respect to $p\Big(\mathbf{x}_{0:T} \,\Big|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}'\Big)$ (assumed implicitly in the notation), applying the identity $\mathbf{a}^\mathsf{T}\mathbf{C}\mathbf{b} = \mathrm{Tr}\Big[\mathbf{a}^\mathsf{T}\mathbf{C}\mathbf{b}\Big] = \mathrm{Tr}\Big[\mathbf{C}\mathbf{b}\mathbf{a}^\mathsf{T}\Big]$, and dropping the constant terms we get

$$\begin{aligned}\mathcal{Q}\Big(\boldsymbol{\theta}, \boldsymbol{\theta}'\Big) \approx -\frac{1}{2}\Bigg\{ &\mathrm{Tr}\Big[\boldsymbol{\Sigma}_0^{-1}\Big\langle\left(\mathbf{x}_0 - \boldsymbol{\mu}_0\right)\left(\mathbf{x}_0 - \boldsymbol{\mu}_0\right)^\mathsf{T}\Big\rangle\Big] + \log\det(\boldsymbol{\Sigma}_0) \\ &+ \mathrm{Tr}\Bigg[\mathbf{Q}^{-1}\sum_{k=1}^{T}\Big\langle\left(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}\right)\left(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}\right)^\mathsf{T}\Big\rangle\Bigg] + T\log\det(\mathbf{Q}) \\ &+ \mathrm{Tr}\Bigg[\mathbf{R}^{-1}\sum_{k=1}^{T}\Big\langle\left(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k\right)\left(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k\right)^\mathsf{T}\Big\rangle\Bigg] + T\log\det(\mathbf{R})\Bigg\}.\end{aligned} \tag{108}$$

Let us denote the quadratic forms inside the traces in Equation (108) with

$$\begin{aligned}\mathbf{I}_1(\boldsymbol{\theta},\boldsymbol{\theta}') &= \Big\langle (\mathbf{x}_0-\boldsymbol{\mu}_0)(\mathbf{x}_0-\boldsymbol{\mu}_0)^\mathsf{T}\Big\rangle \\ &= \int_{\mathcal{X}\times T}(\mathbf{x}_0-\boldsymbol{\mu}_0)(\mathbf{x}_0-\boldsymbol{\mu}_0)^\mathsf{T}\, p\Big(\mathbf{x}_{0:T}\,\Big|\,\mathbf{y}_{0:T},\boldsymbol{\theta}'\Big)\,\mathrm{d}\mathbf{x}_{0:T} \\ &= \int_{\mathcal{X}}(\mathbf{x}_0-\boldsymbol{\mu}_0)(\mathbf{x}_0-\boldsymbol{\mu}_0)^\mathsf{T}\, p\Big(\mathbf{x}_0\,\Big|\,\mathbf{y}_{0:T},\boldsymbol{\theta}'\Big)\,\mathrm{d}\mathbf{x}_0\end{aligned} \tag{109}$$

$$\begin{aligned}\mathbf{I}_2(\boldsymbol{\theta},\boldsymbol{\theta}') = \sum_{k=1}^{T}\iint_{\mathcal{X}}&(\mathbf{x}_k-\mathbf{A}\mathbf{x}_{k-1})(\mathbf{x}_k-\mathbf{A}\mathbf{x}_{k-1})^\mathsf{T} \\ &\times p\Big(\mathbf{x}_k,\mathbf{x}_{k-1}\,\Big|\,\mathbf{y}_{0:T},\boldsymbol{\theta}'\Big)\,\mathrm{d}\mathbf{x}_k\mathrm{d}\mathbf{x}_{k-1}\end{aligned} \tag{110}$$

$$\mathbf{I}_3(\boldsymbol{\theta},\boldsymbol{\theta}') = \sum_{k=1}^{T}\int_{\mathcal{X}}(\mathbf{y}_k-\mathbf{H}\mathbf{x}_k)(\mathbf{y}_k-\mathbf{H}\mathbf{x}_k)^\mathsf{T}\, p\Big(\mathbf{x}_k\,\Big|\,\mathbf{y}_{0:T},\boldsymbol{\theta}'\Big)\,\mathrm{d}\mathbf{x}_k. \tag{111}$$

It is clear then that in the E-step one needs to compute the $T+1$ smoothing distributions, including the T cross-timestep distributions, since these will be needed in the expectations. By applying the identity

$$\mathrm{var}[\mathbf{x}] = \Big\langle \mathbf{x}\mathbf{x}^\mathsf{T}\Big\rangle - \langle\mathbf{x}\rangle\langle\mathbf{x}\rangle^\mathsf{T}, \tag{112}$$

we can write the first expectation as

$$\mathbf{I}_1(\boldsymbol{\theta},\boldsymbol{\theta}') = \mathbf{P}_{0|T} + (\mathbf{m}_{0|T}-\boldsymbol{\mu}_0)(\mathbf{m}_{0|T}-\boldsymbol{\mu}_0)^\mathsf{T}. \tag{113}$$

This was a result of assuming the Gaussian prior distribution of Equation (5c).

As in (39), let us denote the joint smoothing distribution of $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ by

$$p(\mathbf{x}_{k-1},\mathbf{x}_k\,|\,\mathbf{y}_{0:T}) = \mathrm{N}\left(\begin{bmatrix}\mathbf{x}_{k-1}\\ \mathbf{x}_k\end{bmatrix}\,\middle|\,\begin{bmatrix}\mathbf{m}_{k-1|T}\\ \mathbf{m}_{k|T}\end{bmatrix},\begin{bmatrix}\mathbf{P}_{k-1|T} & \mathbf{D}_{k-1}\\ \mathbf{D}_{k-1}^\mathsf{T} & \mathbf{P}_{k|T}\end{bmatrix}\right). \tag{114}$$

Then by applying the manipulation

$$\begin{aligned}&\Big\langle (\mathbf{A}\mathbf{x}_{k-1}-\mathbf{x}_k)(\mathbf{A}\mathbf{x}_{k-1}-\mathbf{x}_k)^\mathsf{T}\Big\rangle \\ &= \begin{bmatrix}\mathbf{A}^\mathsf{T}\\ -\mathbf{I}\end{bmatrix}^\mathsf{T}\left\langle\begin{bmatrix}\mathbf{x}_{k-1}\\ \mathbf{x}_k\end{bmatrix}\begin{bmatrix}\mathbf{x}_{k-1}\\ \mathbf{x}_k\end{bmatrix}^\mathsf{T}\right\rangle\begin{bmatrix}\mathbf{A}^\mathsf{T}\\ -\mathbf{I}\end{bmatrix}\end{aligned} \tag{115}$$

we get

$$\mathbf{I}_2(\boldsymbol{\theta},\boldsymbol{\theta}') = \begin{bmatrix}\mathbf{A}^\mathsf{T}\\ -\mathbf{I}\end{bmatrix}^\mathsf{T} \sum_{k=1}^{T} \left(\begin{bmatrix}\mathbf{P}_{k-1|T} & \mathbf{D}_{k-1}\\ \mathbf{D}_{k-1}^\mathsf{T} & \mathbf{P}_{k|T}\end{bmatrix} + \begin{bmatrix}\mathbf{m}_{k-1|T}\\ \mathbf{m}_{k|T}\end{bmatrix}\begin{bmatrix}\mathbf{m}_{k-1|T}\\ \mathbf{m}_{k|T}\end{bmatrix}^\mathsf{T} \right) \begin{bmatrix}\mathbf{A}^\mathsf{T}\\ -\mathbf{I}\end{bmatrix} \tag{116}$$

$$\begin{aligned}
&= \begin{bmatrix}\mathbf{A}^\mathsf{T}\\ -\mathbf{I}\end{bmatrix}^\mathsf{T} \begin{bmatrix}\sum_{k=1}^{T}\left\langle \mathbf{x}_{k-1}\mathbf{x}_{k-1}^\mathsf{T}\right\rangle & \sum_{k=1}^{T}\left\langle \mathbf{x}_{k-1}\mathbf{x}_{k}^\mathsf{T}\right\rangle\\ \sum_{k=1}^{T}\left\langle \mathbf{x}_{k}\mathbf{x}_{k-1}^\mathsf{T}\right\rangle & \sum_{k=1}^{T}\left\langle \mathbf{x}_{k}\mathbf{x}_{k}^\mathsf{T}\right\rangle\end{bmatrix} \begin{bmatrix}\mathbf{A}^\mathsf{T}\\ -\mathbf{I}\end{bmatrix}\\
&= \begin{bmatrix}\mathbf{A}^\mathsf{T}\\ -\mathbf{I}\end{bmatrix}^\mathsf{T} \begin{bmatrix}\mathbf{X}_{11} & \mathbf{X}_{10}\\ \mathbf{X}_{10}^\mathsf{T} & \mathbf{X}_{00}\end{bmatrix} \begin{bmatrix}\mathbf{A}^\mathsf{T}\\ -\mathbf{I}\end{bmatrix}\\
&= \mathbf{X}_{00} - \mathbf{A}\mathbf{X}_{10} - \mathbf{X}_{10}^\mathsf{T}\mathbf{A}^\mathsf{T} + \mathbf{A}\mathbf{X}_{11}\mathbf{A}^\mathsf{T}\\
&= \left(\mathbf{A} - \mathbf{X}_{10}^\mathsf{T}\mathbf{X}_{11}^{-1}\right)\mathbf{X}_{11}\left(\mathbf{A} - \mathbf{X}_{10}^\mathsf{T}\mathbf{X}_{11}^{-1}\right)^\mathsf{T} + \mathbf{X}_{00} + \mathbf{X}_{10}^\mathsf{T}\mathbf{X}_{11}^{-1}\mathbf{X}_{10}.
\end{aligned} \tag{117}$$

It's easy to see that the extremum of the last line with respect to $\mathbf{A}$ is obtained by setting

$$\mathbf{A}_{j+1} = \mathbf{X}_{10}^\mathsf{T}\mathbf{X}_{11}^{-1}. \tag{118}$$

Analogously for $\mathbf{I}_3(\boldsymbol{\theta},\boldsymbol{\theta}')$ we get

$$\mathbf{I}_3(\boldsymbol{\theta},\boldsymbol{\theta}') = \begin{bmatrix}\mathbf{I}\\ -\mathbf{H}^\mathsf{T}\end{bmatrix}^\mathsf{T} \sum_{k=1}^{T} \begin{bmatrix}\mathbf{y}_k\mathbf{y}_k^\mathsf{T} & \mathbf{y}_k\langle\mathbf{x}_k\rangle^\mathsf{T}\\ \langle\mathbf{x}_k\rangle\mathbf{y}_k^\mathsf{T} & \left\langle\mathbf{x}_k\mathbf{x}_k^\mathsf{T}\right\rangle\end{bmatrix} \begin{bmatrix}\mathbf{I}\\ -\mathbf{H}^\mathsf{T}\end{bmatrix} \tag{119}$$

$$= \begin{bmatrix}\mathbf{I}\\ -\mathbf{H}^\mathsf{T}\end{bmatrix}^\mathsf{T} \sum_{k=1}^{T} \begin{bmatrix}\mathbf{Y}_{00} & \bar{\mathbf{C}}_{00}\\ \bar{\mathbf{C}}_{00}^\mathsf{T} & \mathbf{X}_{00}\end{bmatrix} \begin{bmatrix}\mathbf{I}\\ -\mathbf{H}^\mathsf{T}\end{bmatrix}, \tag{120}$$

giving

$$\mathbf{H}_{j+1} = \bar{\mathbf{C}}_{00}\mathbf{X}_{00}^{-1}. \tag{121}$$

The next task is to derive the M-step maximization equations for the process and measurement model noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$. To achieve this, we will differentiate (108) with respect to these matrices. As can be seen from (108), the terms involving $\mathbf{Q}$ or $\mathbf{R}$ are similar in form and so the resulting maximization equations are analogous. Focusing on $\mathbf{Q}$, it is easier to differentiate with respect to

$\mathbf{Q}^{-1}$:

$$\begin{aligned}\frac{\partial \mathcal{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)}{\partial \mathbf{Q}^{-1}} &= -\frac{1}{2}\frac{\partial}{\partial \mathbf{Q}^{-1}}\operatorname{Tr}\left[\mathbf{Q}^{-1}\sum_{k=1}^{T}\left\langle\left(\mathbf{x}_k - \mathbf{f}_{k-1}\right)\left(\mathbf{x}_k - \mathbf{f}_{k-1}\right)^{\mathsf{T}}\right\rangle\right] \\ &\quad -\frac{T}{2}\frac{\partial}{\partial \mathbf{Q}^{-1}}\log\det\mathbf{Q} \\ &= -\frac{1}{2}\sum_{k=1}^{T}\left\langle\left(\mathbf{x}_k - \mathbf{f}_{k-1}\right)\right)\left(\mathbf{x}_k - \mathbf{f}_{k-1}\right)\right)^{\mathsf{T}}\rangle + \frac{T}{2}\mathbf{Q}, \end{aligned} \tag{122}$$

where we have used Equations (92) and (51) in Petersen and Pedersen, 2008. Setting (122) to zero we get the update equation for the next estimate of $\mathbf{Q}$

$$\begin{aligned}\mathbf{Q}_{j+1} &= \frac{1}{T}\sum_{k=1}^{T}\left\langle\left(\mathbf{x}_k - \mathbf{f}_{k-1}\right)\left(\mathbf{x}_k - \mathbf{f}_{k-1}\right)^{\mathsf{T}}\right\rangle \\ &= \frac{1}{T}\mathbf{I}_2(\boldsymbol{\theta}, \boldsymbol{\theta}'). \end{aligned} \tag{123}$$

The analogous result for $\mathbf{R}$ is given by

$$\begin{aligned}\mathbf{R}_{j+1} &= \frac{1}{T}\sum_{k=1}^{T}\left\langle\left(\mathbf{y}_k - \mathbf{h}_k\right)\left(\mathbf{y}_k - \mathbf{h}_k\right)^{\mathsf{T}}\right\rangle \\ &= \frac{1}{T}\mathbf{I}_3(\boldsymbol{\theta}, \boldsymbol{\theta}'). \end{aligned} \tag{124}$$

All in all, the E-step of the EM algorithm in linear-Gaussian SSMs consists of computing the T joint distributions of Equation (114) with the RTS smoother. After this, the M-step estimates are computed for $\mathbf{Q}$ from Equation (123), for $\mathbf{R}$ from Equation (124), for $\mathbf{A}$ from Equation (118) and for $\mathbf{H}$ from Equation (121).

4.3.3 Nonlinear-Gaussian SSMs

As explained in Section 3.2, in the nonlinear case the filtering and smoothing distributions cannot be computed exactly. Thus the E-step is approximate and the convergence guarantees of EM as an ascent method won't apply anymore. In the fortunate case that the model is linear-in-the-parameters the M-step can be solved in closed form. This situation will be covered later in Section 4.3.4. Currently we will assume however that the model is nonlinear in the parameters as well as in the states so that the simplest form to write the model is given by Equation (5). This situation leads to complications in both the E and the M steps of the EM algorithm. Applying EM to SSMs with partial or approximate E-step is considered at least in

Schön, Wills, and Ninness (2011), Ratna (2008), Doucet, De Freitas, and Gordon (2001), Roweis and Ghahramani (2001), and Goodwin and Aguero (2005).

Our strategy will be to apply Gaussian filtering and smoothing in the E-step to compute the expectations of the sufficient statistics. We will settle for an incremental M-step where we again apply a gradient based optimization method. This leads to the requirement of being able to compute $\nabla_{\boldsymbol{\theta}} \mathcal{Q}\Big(\boldsymbol{\theta}, \boldsymbol{\theta}'\Big)$, that is the gradient of the intermediate quantity with respect to $\boldsymbol{\theta}$. It is quite unclear how many iterations of the optimization algorithm should be run in the M-step since, as pointed out in section 4.3.1, *any* new parameter value that increases the log-likehood suffices. In Lange (1995) a heuristic argument was used to only run a single iteration of Newton's method in the M-step.

Denoting $\mathbf{f}_{k-1} \equiv \mathbf{f}(\mathbf{x}_{k-1}, \boldsymbol{\theta})$ and $\mathbf{h}_k \equiv \mathbf{h}(\mathbf{x}_k, \boldsymbol{\theta})$, we now have

$$\begin{aligned}\mathcal{Q}\Big(\boldsymbol{\theta}, \boldsymbol{\theta}'\Big) \approx -\frac{1}{2}\Bigg\{ &\operatorname{Tr}\Big[\boldsymbol{\Sigma}_0^{-1}\mathbf{I}_1(\boldsymbol{\theta}, \boldsymbol{\theta}')\Big] + \log\det(\boldsymbol{\Sigma}_0) \\ &+ \operatorname{Tr}\Big[\mathbf{Q}^{-1}\widehat{\mathbf{I}}_2(\boldsymbol{\theta}, \boldsymbol{\theta}')\Big] + T\log\det(\mathbf{Q}) \\ &+ \operatorname{Tr}\Big[\mathbf{R}^{-1}\widehat{\mathbf{I}}_3(\boldsymbol{\theta}, \boldsymbol{\theta}')\Big] + T\log\det(\mathbf{R})\Bigg\},\end{aligned} \tag{125}$$

where $\mathbf{I}_1(\boldsymbol{\theta}, \boldsymbol{\theta}')$ was given in Equation (109) and we approximate

$$p\Big(\mathbf{x}_k, \mathbf{x}_{k-1} \,\Big|\, \mathbf{y}_{0:T}, \boldsymbol{\theta}'\Big) \approx \mathrm{N}\left(\begin{bmatrix}\mathbf{x}_{k-1} \\ \mathbf{x}_k\end{bmatrix} \,\middle|\, \begin{bmatrix}\mathbf{m}_{k-1|T} \\ \mathbf{m}_{k|T}\end{bmatrix}, \begin{bmatrix}\mathbf{P}_{k-1|T} & \mathbf{D}_{k-1} \\ \mathbf{D}_{k-1}^{\mathsf{T}} & \mathbf{P}_{k|T}\end{bmatrix}\right) \tag{126}$$

giving

$$\begin{aligned}\widehat{\mathbf{I}}_2(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \sum_{k=1}^{T} \iint_{\mathcal{X}} \left(\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}, \boldsymbol{\theta})\right)\left(\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}, \boldsymbol{\theta})\right)^{\mathsf{T}} \\ &\quad \times \mathrm{N}\left(\begin{bmatrix}\mathbf{x}_{k-1} \\ \mathbf{x}_k\end{bmatrix} \,\middle|\, \begin{bmatrix}\mathbf{m}_{k-1|T} \\ \mathbf{m}_{k|T}\end{bmatrix}, \begin{bmatrix}\mathbf{P}_{k-1|T} & \mathbf{D}_{k-1} \\ \mathbf{D}_{k-1}^{\mathsf{T}} & \mathbf{P}_{k|T}\end{bmatrix}\right) \mathrm{d}\mathbf{x}_{k-1}\,\mathrm{d}\mathbf{x}_k \\ &= \begin{bmatrix}\mathbf{I} \\ -\mathbf{I}\end{bmatrix}^{\mathsf{T}} \sum_{k=1}^{T} \left\langle \begin{bmatrix}\mathbf{x}_k \\ \mathbf{f}_{k-1}\end{bmatrix} \begin{bmatrix}\mathbf{x}_k \\ \mathbf{f}_{k-1}\end{bmatrix}^{\mathsf{T}} \right\rangle \begin{bmatrix}\mathbf{I} \\ -\mathbf{I}\end{bmatrix}\end{aligned} \tag{127}$$

and

$$\begin{aligned}\widehat{\mathbf{I}}_3(\boldsymbol{\theta},\boldsymbol{\theta}') &= \sum_{k=1}^{T}\int_{\mathcal{X}}(\mathbf{y}_k-\mathbf{h}(\mathbf{x}_k,\boldsymbol{\theta}))\,(\mathbf{y}_k-\mathbf{h}(\mathbf{x}_k,\boldsymbol{\theta}))^{\mathsf{T}}\,\mathrm{N}\Big(\mathbf{x}_k\,\Big|\,\mathbf{m}_{k|T},\mathbf{P}_{k|T}\Big)\,\mathrm{d}\mathbf{x}_k \\ &= \begin{bmatrix}\mathbf{I}\\ -\mathbf{I}\end{bmatrix}^{\mathsf{T}}\sum_{k=1}^{T}\left\langle\begin{bmatrix}\mathbf{y}_k\\ \mathbf{h}_k\end{bmatrix}\begin{bmatrix}\mathbf{y}_k\\ \mathbf{h}_k\end{bmatrix}^{\mathsf{T}}\right\rangle\begin{bmatrix}\mathbf{I}\\ -\mathbf{I}\end{bmatrix}.\end{aligned} \tag{128}$$

Clearly the integrals (127) and (128) are Gaussian expectation integrals of the form (40). An obvious strategy is thus to utilize a Gaussian smoother to compute the joint smoothing distributions and then compute the $2T$ expectation integrals by applying the same integration rule as was used by the smoother.

To use gradient based nonlinear optimization in the M-step, we will need the analytical gradient of the objective function. It is important to highlight at this point that the joint smoothing distribution approximation of Equation (126) depends on $\boldsymbol{\theta}'$ (the *current*, e.g. given, parameter value) and during the M-step we are searching for the *next* parameter value $\boldsymbol{\theta}'' = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}\big(\boldsymbol{\theta},\boldsymbol{\theta}'\big)$. In other words when differentiating the integrals (127) and (128) the Gaussian functions are independent of $\boldsymbol{\theta}$. Let us then find out the formal differential of a general log-Gaussian, where both the mean and the variance depend on the scalar parameter θ. We get

$$\begin{aligned}&\frac{\partial}{\partial\theta}\log\mathrm{N}(\mathbf{x}\,|\,\mathbf{m},\mathbf{P}) \\ &\quad= -\frac{1}{2}\frac{\partial}{\partial\theta}\Big[(\mathbf{x}-\mathbf{m})^{\mathsf{T}}\mathbf{P}^{-1}(\mathbf{x}-\mathbf{m})\Big] - \frac{1}{2}\frac{\partial}{\partial\theta}\log\det(\mathbf{P}) \\ &\quad= -\frac{1}{2}\frac{\partial}{\partial\theta}\mathrm{Tr}\Big[\mathbf{P}^{-1}(\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^{\mathsf{T}}\Big] - \frac{1}{2}\mathrm{Tr}\left[\mathbf{P}^{-1}\frac{\partial\mathbf{P}}{\partial\theta}\right] \\ &\quad= \frac{1}{2}\mathrm{Tr}\left[\mathbf{P}^{-1}\left(\frac{\partial\mathbf{P}}{\partial\theta}\,\mathbf{P}^{-1}(\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^{\mathsf{T}} + 2\,\frac{\partial\mathbf{m}}{\partial\theta}\,(\mathbf{x}-\mathbf{m})^{\mathsf{T}} - \frac{\partial\mathbf{P}}{\partial\theta}\right)\right]\end{aligned}$$

If we then assume that $\mathbf{f}(\mathbf{x}_{k-1},\boldsymbol{\theta})$, $\mathbf{Q}$, $\mathbf{h}(\mathbf{x}_k,\boldsymbol{\theta})$, $\mathbf{R}$, $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ depend on $\theta_i \in \boldsymbol{\theta}$, we can write

$$\nabla_{\boldsymbol{\theta}}\mathcal{Q}\big(\boldsymbol{\theta},\boldsymbol{\theta}'\big) = \left[\frac{\partial\mathcal{Q}\big(\boldsymbol{\theta},\boldsymbol{\theta}'\big)}{\partial\theta_1} \quad \cdots \quad \frac{\partial\mathcal{Q}\big(\boldsymbol{\theta},\boldsymbol{\theta}'\big)}{\partial\theta_{d_\theta}}\right]^{\mathsf{T}} \tag{129}$$

and

$$\begin{aligned}\frac{\partial \mathcal{Q}\Big(\boldsymbol{\theta},\boldsymbol{\theta}'\Big)}{\partial \theta_i} \approx \\ \frac{1}{2}\mathrm{Tr}\Bigg[\boldsymbol{\Sigma}_0^{-1}\Bigg(\frac{\partial \boldsymbol{\Sigma}_0}{\partial \theta_i}\,\boldsymbol{\Sigma}_0^{-1}\mathbf{I}_1(\boldsymbol{\theta},\boldsymbol{\theta}') + 2\,\frac{\partial \boldsymbol{\mu}_0}{\partial \theta_i}\Big(\mathbf{m}_{0|T} - \boldsymbol{\mu}_0\Big)^{\mathsf{T}} - \frac{\partial \boldsymbol{\Sigma}_0}{\partial \theta_i}\Bigg)\Bigg] \\ +\frac{1}{2}\mathrm{Tr}\Bigg[\mathbf{Q}^{-1}\Bigg(\frac{\partial \mathbf{Q}}{\partial \theta_i}\,\mathbf{Q}^{-1}\widehat{\mathbf{I}}_2(\boldsymbol{\theta},\boldsymbol{\theta}') + 2\sum_{k=1}^{T}\Big\langle\frac{\partial \mathbf{f}(\mathbf{x}_{k-1},\boldsymbol{\theta})}{\partial \theta_i}\,\mathbf{x}_k^{\mathsf{T}}\Big\rangle \\ - 2\sum_{k=1}^{T}\Big\langle\frac{\partial \mathbf{f}(\mathbf{x}_{k-1},\boldsymbol{\theta})}{\partial \theta_i}\,\mathbf{f}(\mathbf{x}_{k-1},\boldsymbol{\theta})^{\mathsf{T}}\Big\rangle - T\,\frac{\partial \mathbf{Q}}{\partial \theta_i}\Bigg)\Bigg] \\ +\frac{1}{2}\mathrm{Tr}\Bigg[\mathbf{R}^{-1}\Bigg(\frac{\partial \mathbf{R}}{\partial \theta_i}\,\mathbf{R}^{-1}\widehat{\mathbf{I}}_3(\boldsymbol{\theta},\boldsymbol{\theta}') + 2\sum_{k=1}^{T}\Big\langle\frac{\partial \mathbf{h}(\mathbf{x}_k,\boldsymbol{\theta})}{\partial \theta_i}\Big\rangle\mathbf{y}_k^{\mathsf{T}} \\ - 2\sum_{k=1}^{T}\Big\langle\frac{\partial \mathbf{h}(\mathbf{x}_k,\boldsymbol{\theta})}{\partial \theta_i}\,\mathbf{h}(\mathbf{x}_k,\boldsymbol{\theta})^{\mathsf{T}}\Big\rangle - T\,\frac{\partial \mathbf{R}}{\partial \theta_i}\Bigg)\Bigg].\end{aligned} \tag{130}$$

In order to gather the computations needed evaluate $\mathcal{Q}\Big(\boldsymbol{\theta},\boldsymbol{\theta}'\Big)$ and $\nabla_{\boldsymbol{\theta}}\mathcal{Q}\Big(\boldsymbol{\theta},\boldsymbol{\theta}'\Big)$ given the sufficient statistics of the T joint smoothing distributions, let us introduce the shorthand notation $\mathbf{f}_{k-1,i}^{\nabla} \equiv \dfrac{\partial \mathbf{f}(\mathbf{x}_{k-1},\boldsymbol{\theta})}{\partial \theta_i}$ and $\mathbf{h}_{k,i}^{\nabla} \equiv \dfrac{\partial \mathbf{h}(\mathbf{x}_k,\boldsymbol{\theta})}{\partial \theta_i}$. One should then perform the following operations:

1. For $k = 1,\ldots,T$ and $i = 1,\ldots,d_\theta$, apply a numerical integration scheme to compute

$$\Bigg\langle\begin{bmatrix}\mathbf{x}_k\\ \mathbf{f}_{k-1}\\ \mathbf{f}_{k-1,i}^{\nabla}\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{f}_{k-1}\\ \mathbf{f}_{k-1,i}^{\nabla}\end{bmatrix}^{\mathsf{T}}\Bigg\rangle \tag{131}$$

   and

$$\Bigg\langle\begin{bmatrix}\mathbf{y}_k\\ \mathbf{h}_k\\ \mathbf{h}_{k,i}^{\nabla}\end{bmatrix}\begin{bmatrix}\mathbf{y}_k\\ \mathbf{h}_k\\ \mathbf{h}_{k,i}^{\nabla}\end{bmatrix}^{\mathsf{T}}\Bigg\rangle. \tag{132}$$

2. Compute $\mathbf{I}_1(\boldsymbol{\theta},\boldsymbol{\theta}')$, $\widehat{\mathbf{I}}_2(\boldsymbol{\theta},\boldsymbol{\theta}')$, $\widehat{\mathbf{I}}_3(\boldsymbol{\theta},\boldsymbol{\theta}')$ and $\mathcal{Q}\Big(\boldsymbol{\theta},\boldsymbol{\theta}'\Big)$ from Equations (113), (127), (128) and (125) respectively.

3. Compute $\nabla_{\boldsymbol{\theta}}\mathcal{Q}\Big(\boldsymbol{\theta},\boldsymbol{\theta}'\Big)$ from Equation (130).

4.3.4 Score computation

As can be understood from Fisher's identity in Equation (95), the gradient of the intermediate quantity $\nabla_{\boldsymbol{\theta}}\mathcal{Q}\Big(\boldsymbol{\theta},\boldsymbol{\theta}'\Big)$ is equal to the log-likelihood gradient (the score) at the point $\boldsymbol{\theta}=\boldsymbol{\theta}'$. This leads to an alternative computational strategy to the sensitivity equations of Section 4.2.1, termed the *easy gradient recipe* in Olsson, Petersen, and Lehn-Schiøler (2007). Thus to compute the score at $\boldsymbol{\theta}'$ one performs the computations detailed in the previous section for evaluating $\nabla_{\boldsymbol{\theta}}\mathcal{Q}\Big(\boldsymbol{\theta}',\boldsymbol{\theta}'\Big)$. Using gradient based optimization for $\ell\Big(\boldsymbol{\theta}\Big)$, where $\nabla\ell\Big(\boldsymbol{\theta}\Big)$ is computed through the Fisher's identity is also the idea in the expectation-conjugate-gradient (ECG) method of Salakhutdinov, Roweis, and Ghahramani (2003b).

Linear-in-the-parameters SSM:s

If the dynamic and measurement models are linear-in-the-parameters but nonlinear in the states, then only the E-step is approximate and the M-step can be performed in closed form. Thus this situation is a combination of the linear-Gaussian and the nonlinear-Gaussian cases discussed in the previous sections.

Suppose now that $\mathbf{f}:\mathcal{X}\rightarrow\mathcal{X}$ is a linear combination of vector valued functions $\boldsymbol{\rho}_j:\mathcal{X}\rightarrow\mathbb{R}^{d_{\Phi,j}}$, so that the parameters of $\mathbf{f}$, $\boldsymbol{\Phi}_j$, are matrices of size $d_x\times d_{\Phi,j}$. Then $\mathbf{f}$ can be written as

$$\begin{aligned}\mathbf{f}(\mathbf{x}_{k-1},\boldsymbol{\theta}) &= \boldsymbol{\Phi}_1(\boldsymbol{\theta})\boldsymbol{\rho}_1(\mathbf{x}_{k-1})+\cdots+\boldsymbol{\Phi}(\boldsymbol{\theta})_m\boldsymbol{\rho}_m(\mathbf{x}_{k-1})\\ &= \begin{bmatrix}\boldsymbol{\Phi}(\boldsymbol{\theta})_1 & \ldots & \boldsymbol{\Phi}(\boldsymbol{\theta})_m\end{bmatrix}\begin{bmatrix}\boldsymbol{\rho}_1(\mathbf{x}_{k-1})\\ \vdots\\ \boldsymbol{\rho}_m(\mathbf{x}_{k-1})\end{bmatrix}\\ &= \mathbf{A}(\boldsymbol{\theta})\mathbf{g}(\mathbf{x}_{k-1}),\end{aligned} \tag{133}$$

so that $\mathbf{A}(\boldsymbol{\theta})$ is now a matrix of size $d_x\times\sum_{j=1}^m d_{\Phi,j}$ and $\mathbf{g}:\mathcal{X}\rightarrow\mathbb{R}^{\sum_{j=1}^m d_{\Phi,j}}$. Denoting $\mathbf{g}\left(\mathbf{x}_{k-1}\right)\equiv\mathbf{g}_{k-1}$ and following the derivation in Equation (116) we now have

$$\widehat{\mathbf{I}}_2(\boldsymbol{\theta},\boldsymbol{\theta}')=\begin{bmatrix}\mathbf{I}\\ -\mathbf{A}^\mathsf{T}\end{bmatrix}^\mathsf{T}\begin{bmatrix}\sum_{k=1}^T\Big\langle\mathbf{x}_k\mathbf{x}_k^\mathsf{T}\Big\rangle & \sum_{k=1}^T\Big\langle\mathbf{x}_k\mathbf{g}_{k-1}^\mathsf{T}\Big\rangle\\ \sum_{k=1}^T\Big\langle\mathbf{g}_{k-1}\mathbf{x}_k^\mathsf{T}\Big\rangle & \sum_{k=1}^T\Big\langle\mathbf{g}_{k-1}\mathbf{g}_{k-1}^\mathsf{T}\Big\rangle\end{bmatrix}\begin{bmatrix}\mathbf{I}\\ -\mathbf{A}^\mathsf{T}\end{bmatrix} \tag{134}$$

Then similarly to (118)

$$\mathbf{A}_{j+1} = \Big(\sum_{k=1}^{T}\Big\langle \mathbf{x}_k \mathbf{g}_{k-1}^{\mathsf{T}}\Big\rangle\Big)\Big(\sum_{k=1}^{T}\Big\langle \mathbf{g}_{k-1}\mathbf{g}_{k-1}^{\mathsf{T}}\Big\rangle\Big)^{-1}. \tag{135}$$

Analogously for $\mathbf{h}$ we can write

$$\begin{aligned}\mathbf{h}(\mathbf{x}_k, \boldsymbol{\theta}) &= \boldsymbol{\Upsilon}_1(\boldsymbol{\theta})\boldsymbol{\pi}_1(\mathbf{x}_k) + \cdots + \boldsymbol{\Upsilon}_m(\boldsymbol{\theta})\boldsymbol{\pi}_m(\mathbf{x}_k) \\ &= \begin{bmatrix}\boldsymbol{\Upsilon}_1 & \dots & \boldsymbol{\Upsilon}_m\end{bmatrix}\begin{bmatrix}\boldsymbol{\pi}_1(\mathbf{x}_k) \\ \vdots \\ \boldsymbol{\pi}_m(\mathbf{x}_k)\end{bmatrix} \\ &= \mathbf{H}(\boldsymbol{\theta})\mathbf{b}(\mathbf{x}_k),\end{aligned} \tag{136}$$

where $\mathbf{H}(\boldsymbol{\theta})$ is now $d_x \times \sum_{j=1}^{m} d_{\Upsilon,j}$ and $\mathbf{b} : \mathcal{X} \to \mathbb{R}^{\sum_{j=1}^{m} d_{\Upsilon,j}}$. Denoting $\mathbf{b}\left(\mathbf{x}_k\right) \equiv \mathbf{b}_k$, we get

$$\widehat{\mathbf{I}}_3(\boldsymbol{\theta}, \boldsymbol{\theta}') = \begin{bmatrix}\mathbf{I} \\ -\mathbf{H}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}} \sum_{k=1}^{T}\begin{bmatrix}\mathbf{y}_k\mathbf{y}_k^{\mathsf{T}} & \mathbf{y}_k\langle \mathbf{b}_k\rangle^{\mathsf{T}} \\ \langle \mathbf{b}_k\rangle\mathbf{y}_k^{\mathsf{T}} & \left\langle \mathbf{b}_k\mathbf{b}_k^{\mathsf{T}}\right\rangle\end{bmatrix}\begin{bmatrix}\mathbf{I} \\ -\mathbf{H}^{\mathsf{T}}\end{bmatrix}, \tag{137}$$

giving

$$\mathbf{H}_{j+1} = \Big(\mathbf{y}_k\langle \mathbf{b}_k\rangle^{\mathsf{T}}\Big)\Big(\Big\langle \mathbf{b}_k\mathbf{b}_k^{\mathsf{T}}\Big\rangle\Big)^{-1}. \tag{138}$$

5 Results

5.1 Endoathmospheric flight of a ballistic projectile

Let us consider a situation where a ballistic projectile is launched from the ground into the air. We assume that the situation is governed by Newtonian mechanics and that the projectile experiences a constant known gravitational force, directed towards the ground. In addition we assume a *constant* drag force directed orthogonally to the gravitational force. This is clearly an oversimplification, since in a more realistic model the drag force should be proportional to velocity and directed against it. Furthermore the drag force is highly dependent on air density and so on the altitude (Ristic, Arulampalam, & Gordon, 2004). Nevertheless, we make the aforementioned simplification to keep to resulting SSM linear. The modeling error is mitigated slightly by introducing additive white noise to both forces.

We obtain a sequence of range measurements with a radar, so that our data consists of the noisy two dimensional locations of the object as measured at time points $\{t_k\}_{k=1}^{T}$. We assume a constant interval τ between the time points. With these considerations, the system can be cast into a linear-Gaussian SSM.

In continous time and for a single coordinate χ, the dynamics can now be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\chi(t)\\ \dot{\chi}(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}\chi(t)\\ \dot{\chi}(t)\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix} g_\chi + \begin{bmatrix}0\\ 1\end{bmatrix}\beta(t), \tag{139}$$

where $g_\chi < 0$ is the mean of the force and $\beta(t)$ can be considered a white noise process with variance (or spectral density) σ_χ^2. Thus the state contains the position and its first time derivative, the velocity.

To discretize the dynamics, we will apply a simple integration scheme where $\mathbf{x}(t) = \mathbf{x}(t_k)$ when $t \in \left[t_k, t_{k+1}\right)$ (Bar-Shalom et al., 2004). The system will be modeled in two dimensional Cartesian coordinates with two state components for position and two for velocity, giving $d_x = 4$. The state at time k is then

$$\mathbf{x}_k = \begin{bmatrix}\chi_k & \dot{\chi}_k & \gamma_k & \dot{\gamma}_k\end{bmatrix}^{\mathsf{T}} \tag{140}$$

where $\dot{\chi}_k = \left.\dfrac{\mathrm{d}\chi(t)}{\mathrm{d}t}\right|_{t=t_k}$ and analogously for $\dot{\gamma}_k$. The corresponding measurement is

given by

$$\mathbf{y}_k = \begin{bmatrix} \chi_k & \gamma_k \end{bmatrix}^\mathsf{T} + \mathbf{r}_k, \tag{141}$$

where $\mathbf{r}_k$ is a white noise process with variance σ_r^2. The discrete time linear-Gaussian SSM can now be written as

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}\mathbf{x}_{k-1} + \mathbf{u} + \mathbf{q}_{k-1}, & \mathbf{q}_{k-1} &\sim \mathrm{N}(\mathbf{0}, \mathbf{Q}), \\ \mathbf{y}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{r}_k, & \mathbf{r}_k &\sim \mathrm{N}\Big(\mathbf{0}, \sigma_r^2\mathbf{I}\Big), \end{aligned} \tag{142}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & \tau & & \\ & 1 & & \\ & & 1 & \tau \\ & & & 1 \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} 0 \\ g_\chi \\ 0 \\ g_\gamma \end{bmatrix},$$

$$\mathbf{Q} = \begin{bmatrix} {}^1\!/_3\sigma_\chi^2\tau^3 & {}^1\!/_2\sigma_\chi^2\tau^2 & & \\ {}^1\!/_2\sigma_\chi^2\tau^2 & \sigma_\chi^2\tau & & \\ & & {}^1\!/_3\sigma_\gamma^2\tau^3 & {}^1\!/_2\sigma_\gamma^2\tau^2 \\ & & {}^1\!/_2\sigma_\gamma^2\tau^2 & \sigma_\gamma^2\tau \end{bmatrix}, \qquad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Additionally, the initial state $\mathbf{x}_0$ is assumed to be known with $\mathbf{x}_0 = \begin{bmatrix} 0 & \cos(\alpha_0)v_0 & 0 & \sin(\alpha_0)v_0 \end{bmatrix}^\mathsf{T}$.

Figure 4 presents an example trajectory with the hidden state components obtained by simulation and the corresponding simulated noisy measurements. The simulated model was the linear-Gaussian SSM in Equation (142) with the parameter values presented in Table 1.

Table 1: Parameter values used for simulation in Section 5.1

Parameter	Value	Unit	Parameter	Value	Unit
σ_χ	1.2	m	τ	0.01	s
σ_γ	0.8	m	σ_r	2.5	m
g_χ	-1.8	m/s^2	α_0	60	$^\circ$
g_γ	-9.81	m/s^2	v_0	40	m/s

Let us then proceed to estimating some of the parameters by using the noisy measurements as input to the two parameter estimation methods we have been considering. We choose parameters $\boldsymbol{\theta}_\mathrm{B} = \{g_\chi, g_\gamma, \sigma_r\}$, which are the accelerations caused by the drag force and gravitation as well as the measurement noise standard deviation. The true values, that is, the values which were used for generating the

measurements are presented in Table 1. To inspect the effect of the initial guess as well as well as that of the specific measurement dataset, we ran $M = 100$ simulations with the initial estimate for each parameter θ_i picked from the uniform distribution $U[0, 2\theta_i^*]$, where θ_i^* is the true generative value for parameter i given in Table 1. The lengths of the simulated measurent datasets were around $N \approx 1400$ with some variance caused by always stopping the simulation when $\gamma_k < 0$ for some k. For each simulated dataset and the associated initial estimate we ran the EM and the BFGS parameter estimation methods for joint estimation of the three parameters.

In this case one can find closed form expressions for all three parameters in the EM M-step. The M-step equations for linear-Gaussian SSMs, presented in Equations (118), (121), (123) and (124), do not include one for estimating the constant input $\mathbf{u}$, which in this case contains the accelerations. It is not difficult to derive however and for this particular model it reads

$$\mathbf{u}_{j+1} = \frac{1}{2T\tau^2} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \Big(3\big(m_{0|T} - m_{T|T}\big) + \tau\big(2\sum_{k=1}^{T} m_{k|T} + \sum_{k=1}^{T} m_{k-1|T}\big)\Big). \tag{143}$$

The BFGS implementation used was the `fminunc` function included in the MATLAB Optimization Toolbox (The Mathworks Inc. 2012). It is intended for general unconstrained nonlinear optimization and implements other methods in addition to BFGS. To force BFGS, `fminunc` should be called in the following way

```
opt = optimset(@fminunc);
opt.GradObj = 'on';
opt.LargeScale = 'off'; % use BFGS
% lhg = objective function and gradient
% [lh(x),lh'(x)] = lhg(x)
% p0 = initial estimate
p_min = fminunc(@lhg,p0,opt);
```

The likelihood convergence for both methods is presented in Figure 5. It is important to note that the iteration numbers are not directly comparable between the parameter estimation methods so that one shoudn't attempt to draw conclusions on the *relative* convergence rate betweeen the methods based on the convergence plots.

The parameter convergence results are presented in Figure 6, which contains eight separate panels: one per parameter and estimation method and the likelihoods

for both estimation methods. One line is plotted in every panel for every simulated dataset and the panels display their quantities as a function of the iteration number of the estimation method. The convergence profiles show a lot of variability between the methods and between the parameters but the means of the converged estimates seem to agree very well with the generative values in all cases. Also g_γ and σ_r show very little variance in the converged estimate compared to g_χ. In any case, according to the asymptotic theory of the ML/MAP estimates, the variance should go to zero as the amount of data approaches infinity.

Finally, Table 2 presents the averaged final results. Both methods seem to obtain the same results and in fact they agree to the first six decimal places. This is to be expected, since as mentioned earlier, they can be proved to compute the same quantities in the linear-Gaussian case. The fact that the results differ after the sixth decimal place can be explained by differing numerical properties of the two algorithms. In case of g_γ and σ_r, the estimates agree exactly at least to three decimal places with the generative values, whereas g_χ is correct up to two decimal places. Since we are using unbiased estimates, the estimation error could be diminished up to the order of the machine epsilon by simulating more data points . As a conclusion, it seems that in this case the estimation problem was too simple to bring about noticeable differences in the performance of the parameter estimation methods.

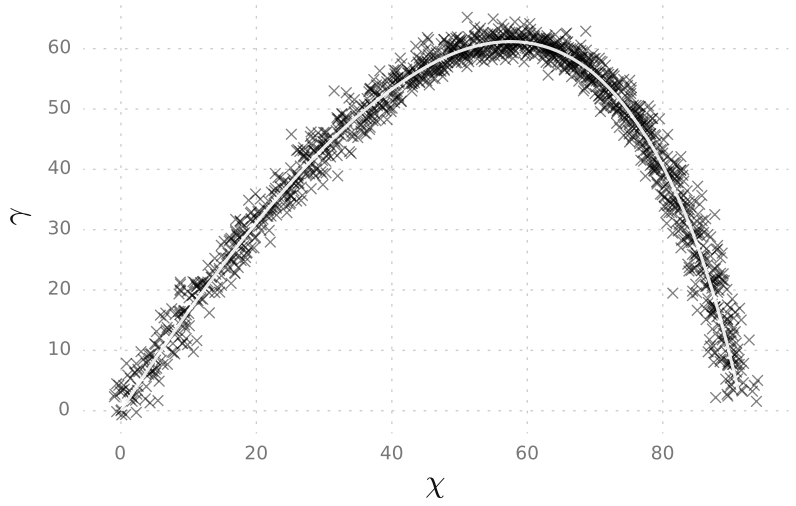

Figure 4: A simulated trajectory (white line) and noisy measurements (black crosses) from the linear-Gaussian SSM (142). The coordinates are in meters. The projectile is simulated for approximately 7 seconds and there are $T = 1396$ measurements.

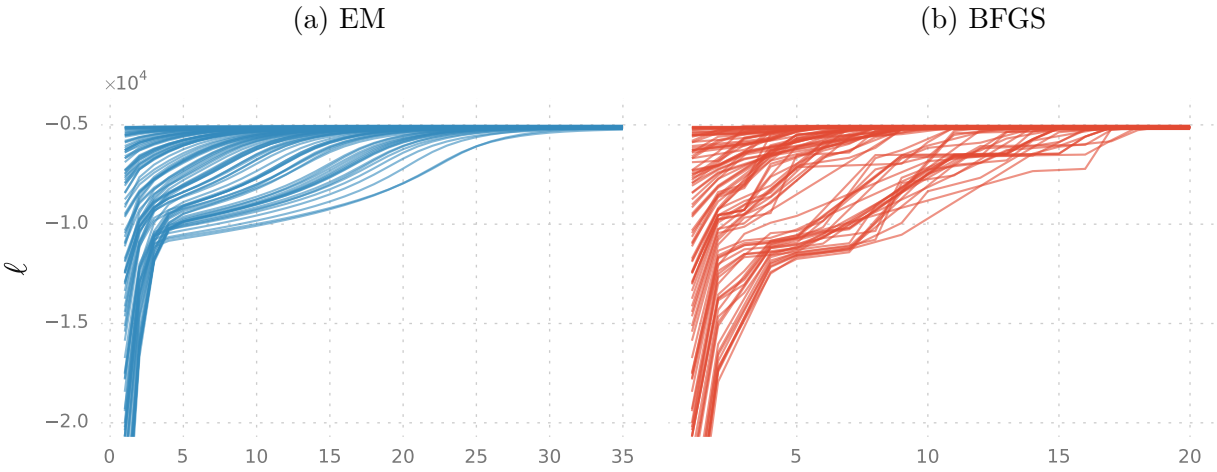

Figure 5: Convergence of the likelihood for $M = 100$ simulated datasets with varying initial parameter estimates. Both EM in (a) and BFGS in (b) converge to the same likelihood value.

Table 2: Estimated parameter values and the final log-likelihood value averaged over 100 simulations in Section 5.1

	g_χ	g_γ	σ_r	$\ell/10^3$
BFGS	−1.796	−9.810	1.500	−5.122
EM	−1.796	−9.810	1.500	−5.122
True	−1.800	−9.810	1.500	

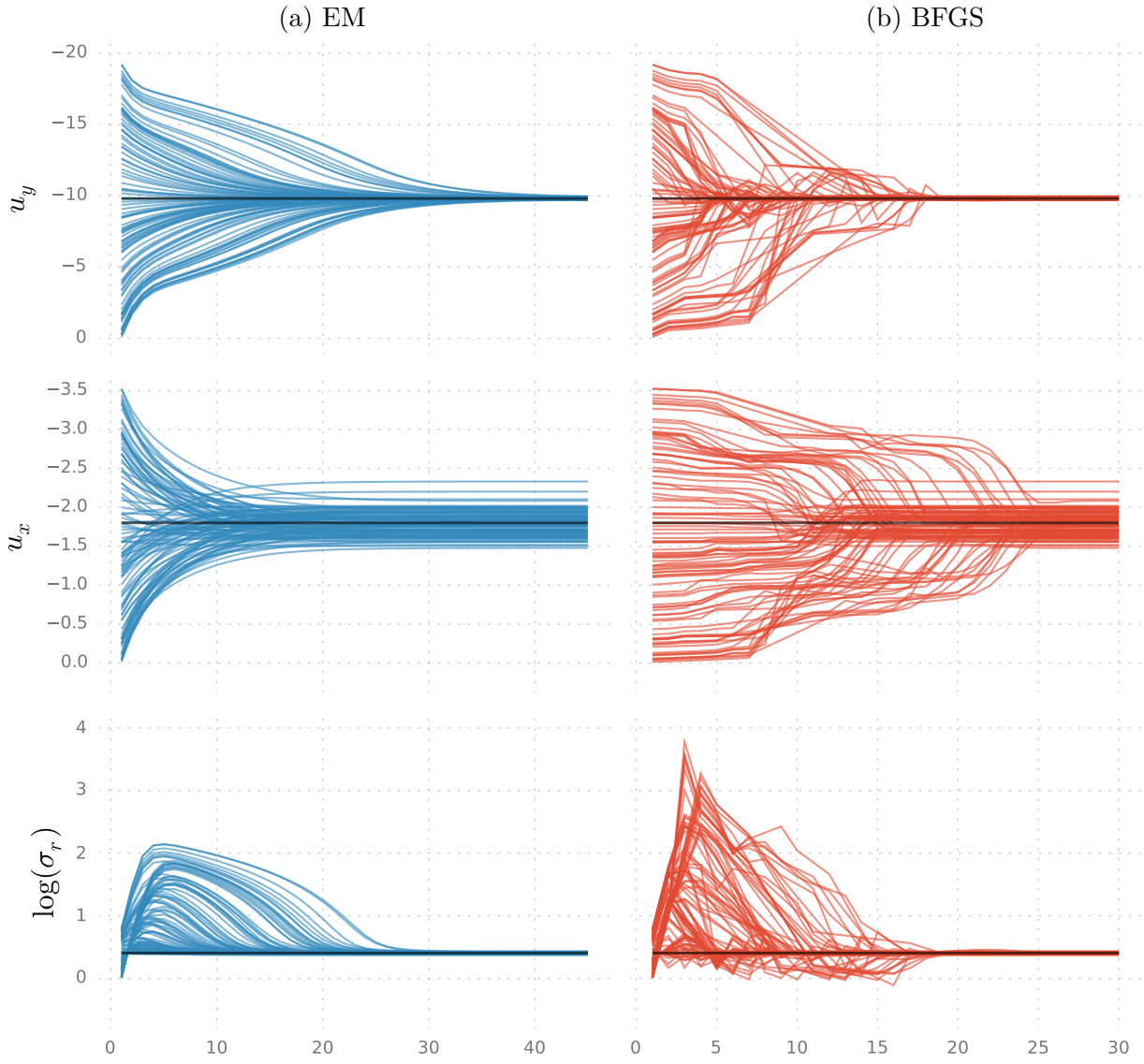

Figure 6: Convergence of the parameter estimates with EM and BFGS as a function of objective function evaluations in Section 5.1. The black line presents the true generative value of the parameter. Note that the objective functions of the optimization methods differ in their computational complexity, implying that the plots cannot be directly compared in the x-axis.

5.2 Photoplethysmograph waveform analysis

The second demonstration is concerned with a nonlinear model for photoplethysmograph (PPG) data, a short sequence of which is presented in Figure 7. PPG is measured with a *pulse oximeter* the functioning of which is based on emitting light (of which the infrared component is relevant to the PPG) through for example a finger or an earlobe. Then either the transmitted or reflected light intensity is measured with a photodiode (Shelley, 2007). As to what exactly is the source of PPG is not without controversy, but as explained by Shelley (2007): "Conceptually, it is most useful to view the pulse oximeter waveform as measuring the change in blood volume (more specifically path length), during a cardiac cycle, in the region being studied (typically the fingertip or earlobe)" (p. 31). The most important use of the PPG is the calculation of arterial oxygen saturation, but it can also be used to estimate the heart rate. In this case a PPG was obtained in connection with a brain imaging study, where a pulse oximeter was attached to the subject while being analysed with fMRI (Särkkä et al., 2012).

A realistic model for this data should take into account the quasi-periodic nature of PPG data, meaning the frequency must be allowed to vary with time. Following the ideas in Särkkä et al. (2012), one possibility is to write the model as a superposition of noisy resonators with time-varying frequencies.

In continous time we can write a stochastic differential equation for the n:th harmonic as

$$\ddot{c}_n(t) = -\omega(t)^2 c_n(t) + \varepsilon_n(t), \tag{144}$$

where $c_n(t)$ is the displacement from equilibrium at time t. The angular velocity ω is related to the frequency f by $\omega(t) = 2\pi f(t)$ and $\varepsilon_n(t)$ is additive white noise with spectral density q_n. For constant frequency and zero spectral density, the solution of Equation (144) is well known to be $c_n(t) = \exp(in\omega t + \phi_n)$, where $\phi_n \in \mathbb{C}$ depends on the initial conditions.

Writing Equation (144) as a vector valued first order differential equation and dividing the noise and the signal derivative by $n\omega(t)$, we get

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} c_n(t) \\ \widehat{c}_n(t) \end{bmatrix} = \begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix}\begin{bmatrix} c_n(t) \\ \widehat{c}_n(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\widehat{\varepsilon}_n(t). \tag{145}$$

As explained in Särkkä et al. (2012), even if Equation (145) is not an exact repre-

sentation of Equation (144), its discretized version has more appealing properties than that of the exact version. Furthermore, the process noise can account for some modeling errors.

Discretizing Equation (145) at equispaced points $\{t_k\}_{k=1}^{T}$ with interval τ and assuming $\omega(t) = \omega(t_k) \equiv \omega_k$ when $t \in [t_k, t_{k+1})$ and that the process noises have equal distributions between the harmonics, we get the following dynamic model for displacement $x^{(n)}$:

$$\begin{bmatrix} x_k^{(n)} \\ \dot{x}_k^{(n)} \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} \cos(n\omega_k) & \sin(n\omega_k) \\ -\sin(n\omega_k) & \cos(n\omega_k) \end{bmatrix} \begin{bmatrix} x_{k-1}^{(n)} \\ \dot{x}_{k-1}^{(n)} \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & \tau\sigma_x^2 \end{bmatrix} \right). \tag{146}$$

We assume that ω_k is part of the state and that its dynamics follow the previously introduced first order random walk model:

$$\omega_k \sim \mathrm{N}(\omega_{k-1}, \tau q_\omega). \tag{147}$$

The joint dynamic model of m harmonics and ω_k is then

$$\underbrace{\begin{bmatrix} \omega_k \\ x_k^{(1)} \\ \dot{x}_k^{(1)} \\ \vdots \\ x_k^{(m)} \\ \dot{x}_k^{(m)} \end{bmatrix}}_{\mathbf{x}_k} = \underbrace{\begin{bmatrix} 1 & & & & & \\ & \cos(\omega_k) & \sin(\omega_k) & & & \\ & -\sin(\omega_k) & \cos(\omega_k) & & & \\ & & & \ddots & & \\ & & & & \cos(m\omega_k) & \sin(m\omega_k) \\ & & & & -\sin(m\omega_k) & \cos(m\omega_k) \end{bmatrix} \begin{bmatrix} \omega_k \\ x_{k-1}^{(1)} \\ \dot{x}_{k-1}^{(1)} \\ \vdots \\ x_{k-1}^{(m)} \\ \dot{x}_{k-1}^{(m)} \end{bmatrix}}_{\mathbf{f}(\mathbf{x}_{k-1})} + \mathbf{q}_{k-1} \tag{148}$$

where

$$\mathbf{q}_{k-1} \sim \mathrm{N}\left(\mathbf{0}, \tau \underbrace{\begin{bmatrix} \sigma_\omega^2 & & & & & \\ & 0 & & & & \\ & & \sigma_x^2 & & & \\ & & & \ddots & & \\ & & & & 0 & \\ & & & & & \sigma_x^2 \end{bmatrix}}_{\mathbf{Q}(\boldsymbol{\theta}_\mathrm{P})} \right). \tag{149}$$

Our objective is now to find the ML estimate (or, equivalently, the MAP estimate with a uniform prior) of the parameter $\boldsymbol{\theta}_{\mathrm{P}} \equiv \{\sigma_\omega, \sigma_x\}$ by using both the gradient based nonlinear optimization approach presented in Section 4.2.2 and the EM approach presented in Section 4.3.3. We will treat the rest of the parameters as fixed with the values presented in Table 3. The first component of the parameter, σ_ω, is the standard deviation of the angular velocity and the second, σ_x, is the standard deviation of the displacement x, shared between the $m = 3$ harmonic components. It would be quite difficult to try to estimate these values based only on a priori information, in contrast to σ_r which could be obtained from the measurement device (the pulse oximeter).

Since $\mathbf{Q}(\boldsymbol{\theta}_{\mathrm{P}})$ is diagonal, we can use Equation (123) in the EM M-step and pick the corresponding elements from the resulting full matrix as our next estimates. Similarly to the analysis of the ballistic projectile in Section 5.1, the BFGS implementation used was the `fminunc` function included in the MATLAB Optimization Toolbox. The CKF filter and CKS smoother of Section 3.2.3 were used as the approximate filtering and smoothing methods respectively. The score function for BFGS was computed by the recursive sensitivity equations of Section 4.2.2.

To analyze the results' sensitivity to the initial estimate, we ran $M = 100$ optimizations with both methods with the initial estimates drawn from a uniform distribution on a suitable interval (that the initial estimate was always the same between the methods). The likelihood convergence for both methods is presented in Figure 8. We note again that the iteration numbers are not directly comparable. As can be seen, the EM estimates seem to converge to values producing identical log likelihood values (on the figure's scale) whereas the BFGS estimates have at least two modes. In contrast to the linear SSM analyzed in the previous section, here we can see that the convergence of the EM is not monotonic in all cases.

The parameter convergence results are presented in Figure 9, which contains four separate panels: one per parameter and estimation method. One line is plotted in every panel for every optimization run and the panels display their quantities as a function of the iteration number of the estimation method. The first thing to note is the vast difference in the behavior of the methods. The EM estimates are quite predictable and either do not converge at all (in the case of σ_w) or converge to the same value (in the case σ_x). The BFGS estimates on the other hand seem to have multiple convergent values for both parameters, depending on the initial estimate.

To get more insight into the behavior of the parameter estimation methods, illustration of the converge of ℓ as a function of the logarithms of both σ_ω and σ_x

is presented in Figure 10. The end points of the runs are marked with black stars. It seems that there are minor local maximums on either side of the major local maximum around $\log \sigma_x \approx -3$. Some of the BFGS optimization runs (31 out of 100 to be specific) converge to the minor local maximums. However, the BFGS runs that do not converge to the minor local maximums converge in both paramaters while the EM runs only converge in $\log \sigma_x$. Since ℓ is very insensitive to changes in σ_ω (at least on the range explored), the variance in the final values for σ_ω for the EM runs has only a negligible effect in ℓ.

Table 4 presents the averaged final results. We have included two sets of estimates for BFGS: one set for all 100 runs and another one averaged over the 69 runs that converge to the major local maximum. The standard errors in BFGS(100) are relatively enormous as expected. However, the standard errors of BFGS(69) are markedly *smaller* across the range when compared to EM(100). A probable explanation for this is the the inconvergence of σ_ω in EM(100).

Table 3: Parameter values used in the PPG analysis in Section 5.2

Parameter	Value	Unit	Parameter	Value	Unit
m	3	–	τ	0.008	s
σ_r	0.001	V			

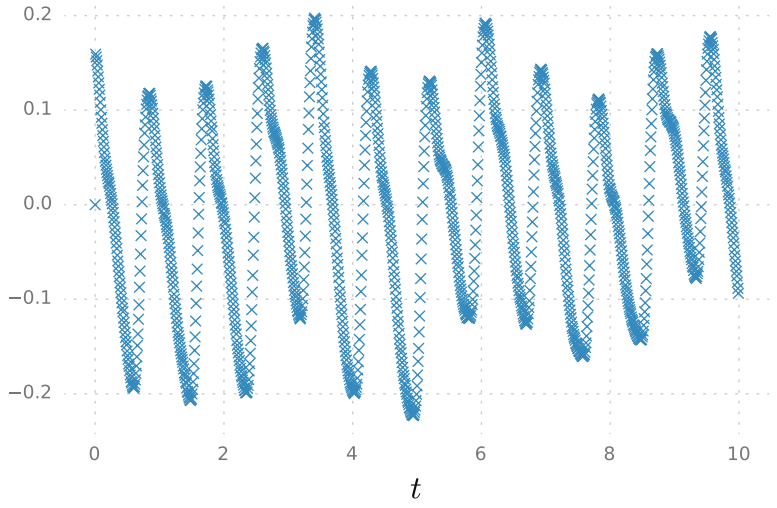

Figure 7: A short sequence of the PPG data in Section 5.2.

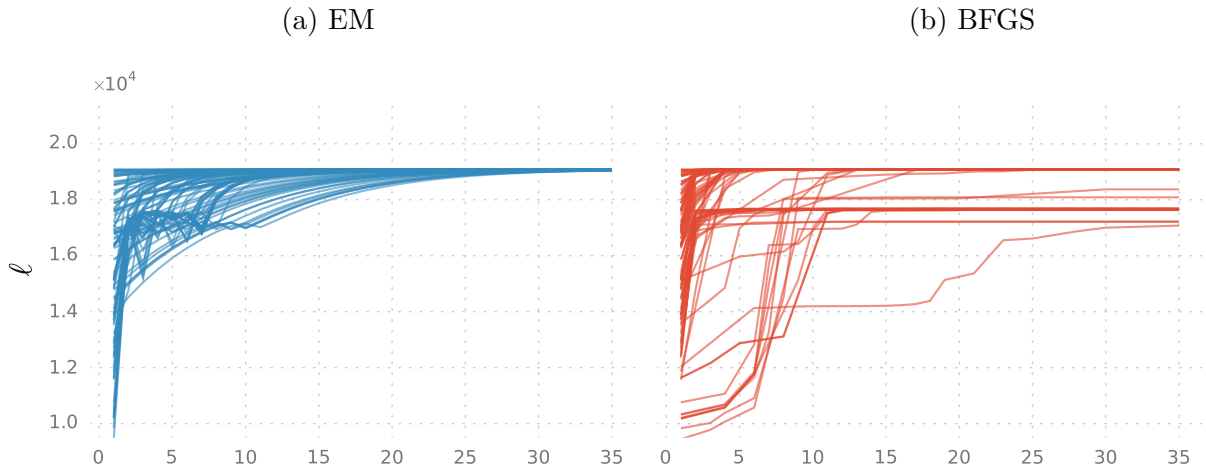

Figure 8: Convergence of the likelihood for $M = 100$ simulated datasets with varying initial parameter estimates in the photoplethysmograph waveform analysis of Section 5.2.

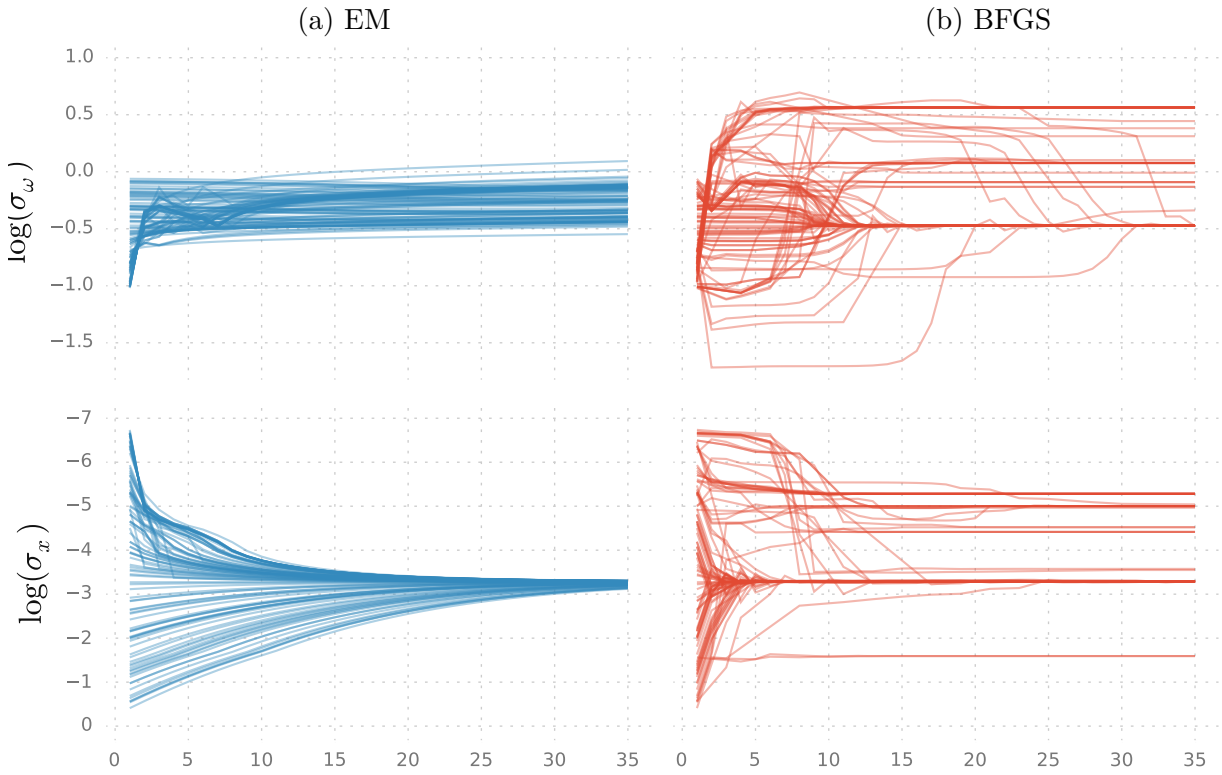

Figure 9: Convergence of the log-parameter estimates with EM (column (a)) and BFGS (column (b)) as a function of objective function evaluations in Section 5.2. Note that the objective functions of the optimization methods differ in their computational complexity, implying that the plots cannot be directly compared in the x-axis.

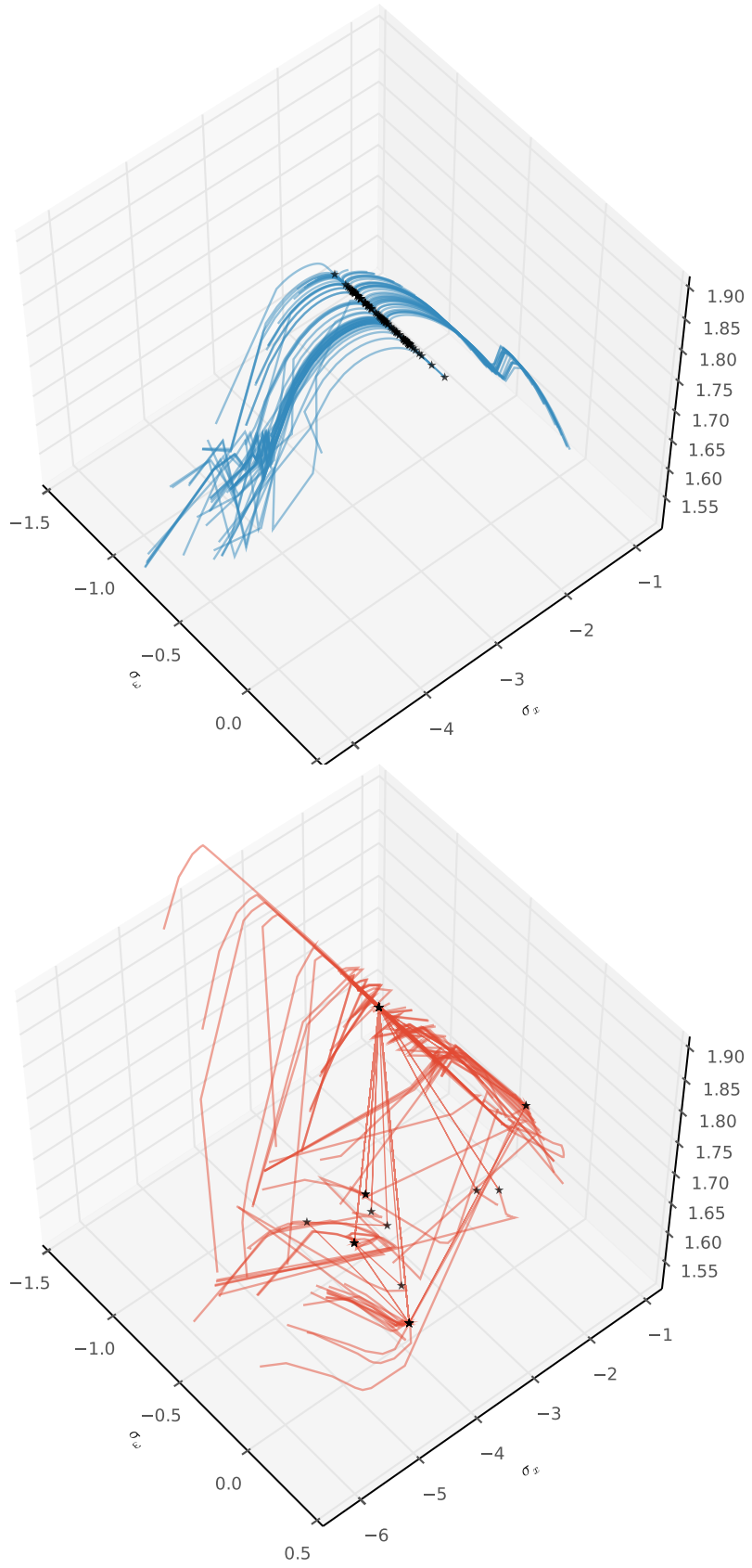

Figure 10: Illustrations of the converge of the log-likelihood as a function of the logarithms of both σ_ω and σ_x for EM (top) and BFGS (bottom). There are 100 independent optimization runs per method with differing initial estimates, so that equal initial estimates were used between the methods. The end points of the runs are marked with black stars.

Table 4: Estimated final parameter and log-likelihood values in Section 5.2. The value in parentheses is the amount of independent optimization runs averaged over. The ± columns are the standard errors of the estimates in the preceding columns. BFGS(69) is included, since some runs in BFGS(100) converge to minor local maximums of ℓ.

	σ_ω	±	σ_x	±	$\ell/10^4$	±
EM(100)	0.867	1.4×10^{-2}	0.038	7.7×10^{-6}	1.908	9.0×10^{-2}
BFGS(100)	0.844	3.9×10^{-2}	0.032	2.8×10^{-3}	1.864	6.7×10^{1}
BFGS(69)	0.623	4.2×10^{-6}	0.037	1.6×10^{-8}	1.908	2.4×10^{-8}

6 Conclusion

Our aim in this thesis has been to explore the problem of static parameter estimation in both linear-Gaussian and nonlinear-Gaussian SSMs. The chosen approach was to focus on two methods for finding MAP/ML estimates, namely gradient based nonlinear optimization and EM. Since the static parameter estimation problem is tightly coupled with the filtering and smoothing problems, the focus of the first part of the thesis was on state estimation. Nonlinear filtering and smoothing is a considerable problem, where closed form solutions exist only in very few situations. We advocated the Gaussian filtering approach and more specifically the cubature Kalman filter and smoother. If the filtering and/or smoothing distributions are well approximated by a Gaussian, these methods offer good approximate solutions for a fraction of the computational complexity of the more general simulation based SMC methods.

The parameter estimation methods we have considered have specific strengths and weaknesses which make them recommendable depending on the model. Let us go through some of these and point out when they are evident in the two demonstrations of Section 5.

Gradient based nonlinear optimization

An important difference in the gradient based nonlinear optimization approach when compared to EM is that the smoothing distributions are not needed. Neither does one need to figure out the model-dependent M-step maximization equations. The marginal likelihood (Equation (62)), or an approximation to it if the model is nonlinear (Equation (72)), is obtained directly from the filtering algorithm.

A number of efficient gradient based nonlinear optimization algorithms are available. We focused on the quasi-Newton BFGS algorithm, which is implemented in MATLAB's `fminunc`. Another BFGS implementation is `ucminf` which has a MATLAB version as well as a version for the open source R software environment (Nielsen, 2000; Nielsen & Mortensen, 2012; R Core Team, 2012). As described in Nielsen (2000), implementing a robust quasi-Newton method is far from straightforward and if the objective is ML/MAP parameter estimation of SSMs, it makes a lot of sense the utilize an off the shelf algorithm. The main appeal of the gradient based nonlinear optimization methods is their order of convergence, which can be quadratic.

The main issue is the score function computation. There are two alternative

routes: either the sensitivity equations, described in Section 4.2.1 for the linear case and in Section 4.2.2 for the nonlinear, or using Fisher's identity and the EM machinery as described in Section 4.3.4. Using the sensitivity equations leads to recursive equations which need to be run alongside and tightly coupled to the chosen filtering algorithm. This is an issue from the perspective of being able to use decoupled modular algorithms. Moreover, the Jacobian matrices of $\mathbf{f}(\mathbf{x}_{k-1}, \boldsymbol{\theta})$ and $\mathbf{h}(\mathbf{x}_k, \boldsymbol{\theta})$ are required. As for the computational complexity, the sensitivity equations scale as d_θ Kalman filters (Cappé et al., 2005; Olsson et al., 2007). A Kalman filter scales as $\mathcal{O}(n^3)$.

With Fisher's identity the parameter estimation algorithm can be made less coupled to the filtering algorithm, but then a separate smoothing step is required. This option doesn't require the Jacobians and is is better suited for modular implementation. Moreover, since the smoothing algorithm can be considered to have the same computational complexity as a Kalman filter, using Fisher's identity has approximately the computational complexity of two Kalman filters.

EM

Using Fisher's identity as part of a nonlinear gradient based optimization method can be considered to be some sort of hybrid approach between the sensitivity equations and the full EM solution. Thus some of the strengths and weaknesses that were mentioned in the previous chapter when comparing sensitivity equations and Fisher's identity apply directly when comparing the sensitivity equations to the EM solution.

A critical question in this comparison is whether the M-step maximization equations can be computed in closed form. If this is not the case and the M-step includes some sort of gradient based nonlinear optimization in itself as part of a generalized EM (gEM) method (mentioned in section 4.3.1) using EM certainly loses some of its appeal.

When the M-step maximization equations can be computed in closed form, EM gains some strengths. First and foremost, no gradient computations are needed. As a consequence, in this case EM is also independent of the parameterization, since the M-step consists only of maximization operations (Cappé et al., 2005). This is not true for the gradient based methods.

As for the order of convergence of EM, there exists some rather interesting results which are discussed at least in Salakhutdinov, Roweis, and Ghahramani (2003a, 2004), Petersen and Winther (2005), and Gibson and Ninness (2005). As a summary,

it seems that the convergence properties depend on the proportion of the total information that is contained in the latent variables. Intuitively, the larger this proportion is, the slower is the convergence. Also, if this proportion is small, the order of convergence can approach that of a true Newton's method (i.e. quadratic). Interesting results concerning the convergence properties of EM (in linear models) were also obtained in Petersen and Winther (2005) and Petersen, Winther, and Hansen (2005). Their analyses seem to show that the order of convergence depends on the signal-to-noise ratio (SNR) in such as a way that the convergence slows down when SNR becomes high. Strategies for speeding up the convergence of EM has been a subject of much interest and some approaches are presented at least in Meng and van Dyk (1997) and Lange (1995).

In the nonlinear case, one has to resort to approximate filtering and smoothing. This means that the marginal log-likelihood and score function computations become approximations. At least with Gaussian filtering and smoothing, these approximations seem to be different between the sensitivity equations and EM (or Fisher's identity). This can be deduced from the fact that when computing the score function through Fisher's identity or the maximization equations of the M-step in EM, we need the quantity $\mathbf{P}_{k-1,k}$ in Equation (29). Since $\mathbf{P}_{k-1,k}$ is part of the Gaussian approximation and it is not needed when approximating the score function through the sensitivity equations, the score function approximations are most probably unequal.

The unequality of the approximations appears to be demonstrated in the results of the two demonstrations in Section 5. With the linear-Gaussian SSM of Section 5.1 the two methods give identical results (when attributing the tiny differences to differing numerical properties). However with the nonlinear-Gaussian SSM of Section 5.2 the results were markedly different, even though both methods were given the same data and initial estimates. Deriving the quantitative difference in the approximations would be an interesting subject for future work.

A Additional material

A.1 Properties of the Gaussian distribution

Lemma A.1. *Suppose* $\mathbf{x} \in \mathbb{R}^n$ *and* $\mathbf{y} \in \mathbb{R}^m$ *have the distributions*

$$p(\mathbf{x}) = \mathrm{N}(\mathbf{x} \,|\, \mathbf{m}, \mathbf{P})$$
$$p(\mathbf{y} \,|\, \mathbf{x}) = \mathrm{N}(\mathbf{y} \,|\, \mathbf{H}\mathbf{x} + \mathbf{u}, \mathbf{R}).$$

Then the joint distribution is

$$p\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}\right) = \mathrm{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \middle| \begin{bmatrix}\mathbf{m}\\ \mathbf{H}\mathbf{m}+\mathbf{u}\end{bmatrix}, \begin{bmatrix}\mathbf{P} & \mathbf{P}\mathbf{H}^\mathsf{T}\\ \mathbf{H}\mathbf{P} & \mathbf{H}\mathbf{P}\mathbf{H}^\mathsf{T}+\mathbf{R}\end{bmatrix}\right)$$

Proof.

$$\begin{aligned}
\langle \mathbf{y} \rangle &= \int \mathbf{y}\, p(\mathbf{y})\, \mathrm{d}\mathbf{y} \\
&= \int \mathbf{y} \Big(\int p(\mathbf{y} \,|\, \mathbf{x}) p(\mathbf{x})\, \mathrm{d}\mathbf{x} \Big)\, \mathrm{d}\mathbf{y} \\
&\quad \text{(change the order of integration according to Fubini's theorem)} \\
&= \Big\langle \langle \mathbf{y} \mid \mathbf{x} \rangle \Big\rangle && \text{(A.1)} \\
&= \mathbf{H}\mathbf{m} + \mathbf{u} && \text{(A.2)} \\
\mathrm{var}[\mathbf{y}] &= \iint (\mathbf{y} - \langle \mathbf{y} \rangle)(\mathbf{y} - \langle \mathbf{y} \rangle)^\mathsf{T} p(\mathbf{y} \,|\, \mathbf{x}) p(\mathbf{x})\, \mathrm{d}\mathbf{y}\mathrm{d}\mathbf{x} \\
&= \Big\langle \langle \mathbf{y} \mid \mathbf{x} \rangle \langle \mathbf{y} \mid \mathbf{x} \rangle^\mathsf{T} \Big\rangle - \Big\langle \langle \mathbf{y} \mid \mathbf{x} \rangle \Big\rangle \Big\langle \langle \mathbf{y} \mid \mathbf{x} \rangle \Big\rangle^\mathsf{T} \\
&\quad + \iint \Big(\mathbf{y} - \langle \mathbf{y} \mid \mathbf{x} \rangle \Big) \Big(\mathbf{y} - \langle \mathbf{y} \mid \mathbf{x} \rangle \Big)^\mathsf{T} p(\mathbf{y} \,|\, \mathbf{x}) p(\mathbf{x})\, \mathrm{d}\mathbf{y}\mathrm{d}\mathbf{x} \\
&= \mathrm{var}\Big[\langle \mathbf{y} \mid \mathbf{x} \rangle \Big] + \Big\langle \mathrm{var}[\mathbf{y} \mid \mathbf{x}] \Big\rangle && \text{(A.3)} \\
&= \mathbf{H}\mathbf{P}\mathbf{H}^\mathsf{T} + \mathbf{R} && \text{(A.4)} \\
\mathrm{cov}[\mathbf{x}, \mathbf{y}] &= \iint \Big(\mathbf{x} - \langle \mathbf{x} \rangle \Big) \Big(\mathbf{y} - \langle \mathbf{y} \rangle \Big)^\mathsf{T} p(\mathbf{x}) p(\mathbf{y} \,|\, \mathbf{x})\, \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \\
&= \int \Big(\mathbf{x} - \langle \mathbf{x} \rangle \Big) \Big(\langle \mathbf{y} \mid \mathbf{x} \rangle - \langle \mathbf{y} \rangle \Big)^\mathsf{T} p(\mathbf{x})\, \mathrm{d}\mathbf{x} \\
&= \int \Big(\mathbf{x} - \mathbf{m} \Big) \Big(\mathbf{x} - \mathbf{m} \Big)^\mathsf{T} \mathbf{H}^\mathsf{T} p(\mathbf{x})\, \mathrm{d}\mathbf{x} \\
&= \mathbf{P}\mathbf{H}^\mathsf{T} && \text{(A.5)}
\end{aligned}$$

□

Lemma A.2. *Suppose* $\mathbf{x} \in \mathbb{R}^n$ *and* $\mathbf{y} \in \mathbb{R}^m$ *have the joint distribution*

$$p\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}\right) = \mathrm{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \,\middle|\, \begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}\mathbf{A} & \mathbf{C}\\ \mathbf{C}^\mathsf{T} & \mathbf{B}\end{bmatrix}\right).$$

Then the marginal and conditional distributions are given by

$$\begin{aligned}
p(\mathbf{x}) &= \mathrm{N}(\mathbf{x} \,|\, \mathbf{a}, \mathbf{A})\\
p(\mathbf{y}) &= \mathrm{N}(\mathbf{x} \,|\, \mathbf{b}, \mathbf{B})\\
p(\mathbf{x} \,|\, \mathbf{y}) &= \mathrm{N}(\mathbf{x} \,|\, \mathbf{a} + \mathbf{C}\mathbf{B}^{-1}(\mathbf{y} - \mathbf{b}), \mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^\mathsf{T})\\
p(\mathbf{y} \,|\, \mathbf{x}) &= \mathrm{N}(\mathbf{x} \,|\, \mathbf{b} + \mathbf{C}^\mathsf{T}\mathbf{A}^{-1}(\mathbf{x} - \mathbf{a}), \mathbf{B} - \mathbf{C}^\mathsf{T}\mathbf{A}^{-1}\mathbf{C})
\end{aligned}$$

References

Anderson, B. D. O., & Moore, J. B. (1979). *Optimal filtering*. Prentice-Hall information and system sciences series. Prentice-Hall.

Andrieu, C., Doucet, A., & Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(3), 269–342.

Arasaratnam, I., & Haykin, S. (2009, June). Cubature Kalman filters. *IEEE Transactions on Automatic Control*, *54*(6), 1254–1269. doi:10.1109/TAC.2009.2019800

Arasaratnam, I., & Haykin, S. (2011, August). Cubature Kalman smoothers. *Automatica*, *47*(10), 2245–2250. doi:10.1016/j.automatica.2011.08.005

Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press.

Barber, D., Cemgil, A. T., & Chiappa, S. (2011). *Inference and estimation in probabilistic time series models*. Cambridge University Press.

Bar-Shalom, Y., Li, X. R., & Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons.

Battiti, R. (1992, March). First- and second-order methods for learning: between steepest descent and Newton's method. *Neural Computation*, *4*(2), 141–166. doi:10.1162/neco.1992.4.2.141

Bernardo, J., Bayarri, M., Berger, J., Beal, M. J., & Ghahramani, Z. (2003). The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. In *Bayesian statistics 7: proceedings of the seventh valencia international meeting* (Vol. 7, pp. 453–464). Oxford University Press.

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer Verlag.

Broyden, C. G., Dennis, J. E., & Moré, J. J. (1973). On the local and superlinear convergence of quasi-Newton methods. *IMA Journal of Applied Mathematics*, *12*(3), 223–245. doi:10.1093/imamat/12.3.223

Cappé, O., Godsill, S. J., & Moulines, E. (2007, May). An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, *95*(5), 899–924. doi:10.1109/JPROC.2007.893250

Cappé, O., Moulines, E., & Rydén, T. (2005). *Inference in hidden Markov models*. Springer Verlag.

Dempster, A., & Laird, N. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. 39*(1), 1–38.

Doucet, A., De Freitas, N., & Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. Statistics for Engineering and Information Science. Springer.

Durbin, J., & Koopman, S. J. (2012). *Time series analysis by state space methods: second edition*. Oxford Statistical Science Series. OUP Oxford.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2004). *Bayesian data analysis*. Chapman & Hall/CRC.

Ghahramani, Z. (1996). *Parameter estimation for linear dynamical systems* (tech. rep. No. CRG-TR-96-2). University of Toronto. Retrieved February 26, 2012, from mlg.eng.cam.ac.uk/zoubin/papers/tr-96-2.pdf

Gibson, S., & Ninness, B. (2005, October). Robust maximum-likelihood estimation of multivariable dynamic systems. *Automatica*, *41*(10), 1667–1682. doi:10.1016/j.automatica.2005.05.008

Godsill, S. J., Vermaak, J., Ng, W., & Li, J. F. (2007). Models and algorithms for tracking of maneuvering objects using variable rate particle filters. *Proceedings of the IEEE*, *95*(5), 925–952. doi:10.1109/JPROC.2007.894708

Goodwin, G., & Aguero, J. (2005, December). Approximate em algorithms for parameter and state estimation in nonlinear stochastic models. In *Decision and control, 2005 and 2005 european control conference. cdc-ecc '05. 44th ieee conference on* (pp. 368 –373). doi:10.1109/CDC.2005.1582183

Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, *140*(2), 107–113.

Grewal, M. S., & Andrews, A. P. (2008). *Kalman filtering: theory and practice using MATLAB*. Wiley.

Gupta, N., & Mehra, R. (1974, December). Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations. *IEEE Transactions on Automatic Control*, *19*(6), 774–783. doi:10.1109/TAC.1974.1100714

Harvey, A. C. (1990). Estimation procedures for structural time series models. *Journal of Forecasting*, *9*(June 1988), 89–108.

Ionides, E., Bhadra, A., Atchadé, Y., & King, A. (2011). Iterated filtering. *The Annals of Statistics*, 1–27. arXiv: arXiv:0902.0347v1

Ito, K. (2000). Gaussian filters for nonlinear filtering problems. *Automatic Control, IEEE Transactions on*, *45*(5), 910–927.

Jazwinski, A. H. (1970). *Stochastic processes and filtering theory*. Mathematics in Science and Engineering. Academic Press.

Jia, B., Xin, M., & Cheng, Y. (2012, February). Sparse-grid quadrature nonlinear filtering. *Automatica*, *48*(2), 327–341. doi:10.1016/j.automatica.2011.08.057

Julier, S., & Uhlmann, J. (1997). A new extension of the Kalman filter to nonlinear systems. In *Int. symp. aerospace/defense sensing, simul. and controls* (Vol. 3, p. 26). Spie Bellingham, WA.

Julier, S., Uhlmann, J., & Durrant-Whyte, H. (2000, March). A new method for the nonlinear transformation of means and covariances in filters and estimators. *Automatic Control, IEEE Transactions on*, *45*(3), 477–482. doi:10.1109/9.847726

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

Kantas, N., Doucet, A., & Singh, S. (2009). An overview of sequential Monte Carlo methods for parameter estimation in general state-space models. *Proceedings of the IFAC*, (1050).

Kushner, H. (1967). Approximations to optimal nonlinear filters. *Automatic Control, IEEE Transactions on*, *12*(5), 546–556. doi:10.1109/TAC.1967.1098671

Lange, K. (1995). A gradient algorithm locally equivalent to the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *57*(2), pp. 425–437.

Ljung, L., & Glad, T. (1994). *Modeling of dynamic systems*. Prentice Hall Information and System Sciences Series. PTR Prentice Hall.

Luenberger, D. G., & Ye, Y. (2008). *Linear and nonlinear programming* (3.). International Series in Operations Research & Management Science. Springer.

MacKay, D. (2003). *Information theory, inference and learning algorithms*. Cambridge University Press.

Mbalawata, I., Särkkä, S., & Haario, H. (2012). Parameter estimation in stochastic differential equations with Markov chain Monte Carlo and non-linear Kalman filtering. *Computational Statistics*, 1–29. Advance online publication. doi:10.1007/s00180-012-0352-y

Meng, X.-L., & van Dyk, D. (1997). The EM algorithm–an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society. Series B (Methodological)*, *59*(3), pp. 511–567.

Merwe, R. V. D. (2004). *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models* (Doctoral dissertation, Oregon Health & Science University).

Murphy, K. (2002, July). *Dynamic Bayesian networks: Representation, inference and learning* (Doctoral dissertation, UC Berkeley, Computer Science Division).

Neal, R., & Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse and other variants. In M. Jordan (Ed.), *Learning in graphical models* (pp. 355–368). Mit Press.

Nielsen, H. B. (2000). *Ucminf – an algorithm for unconstrained, nonlinear optimization* (tech. rep. No. IMM-REP-2000-19). Department of Mathematical Modelling, Technical University of Denmark. Retrieved October 16, 2012, from http://www.imm.dtu.dk/~hbn/publ/TR0019.ps

Nielsen, H. B., & Mortensen, S. B. (2012). Package 'ucminf' [Reference manual]. The Comprehensive R Archive Network. Retrieved October 16, 2012, from http://cran.r-project.org/web/packages/ucminf

Olsson, R. K., Petersen, K. B., & Lehn-Schiøler, T. (2007, April). State-space models: from the EM algorithm to a gradient approach. *Neural Computation*, *19*(4), 1097–1111. doi:10.1162/neco.2007.19.4.1097

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann Publishers.

Petersen, K. B., & Winther, O. (2005). *Explaining slow convergence of EM in low noise linear mixtures*. Technical University of Denmark. Retrieved July 19,

2012, from http://orbit.dtu.dk/getResource?recordId=185954&objectId=1&versionId=1

Petersen, K. B., Winther, O., & Hansen, L. K. (2005, September). On the slow convergence of EM and VBEM in low-noise linear models. *Neural computation*, *17*(9), 1921–1926.

Petersen, K. B., & Pedersen, M. S. (2008, October). The matrix cookbook. Technical University of Denmark. Retrieved October 20, 2012, from http://www2.imm.dtu.dk/pubdb/p.php?3274

Ratna, G. (2008, July). Identification of nonlinear processes with known model structure under missing observations. In C. Myung (Ed.), *Proceedings of the IFAC 17th world congress* (Vol. 11). Seoul, Korea. doi:10.3182/20080706-5-KR-1001.01092

Rauch, H. E., Tung, F., & Striebel, C. T. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, *3*(8), 1445–1450. doi:10.2514/3.3166

R Core Team. (2012). R: A language and environment for statistical computing. Vienna, Austria. Retrieved October 16, 2012, from http://www.r-project.org

Ristic, B., Arulampalam, S., & Gordon, N. (2004). *Beyond the Kalman filter: particle filters for tracking applications*. Artech House Radar Library. Artech House.

Roweis, S. T., & Ghahramani, Z. (2001). Learning nonlinear dynamical system using the expectation-maximization algorithm. In S. Haykin (Ed.), *Kalman filtering and neural networks* (pp. 175–216). Wiley Online Library.

Salakhutdinov, R., Roweis, S., & Ghahramani, Z. (2003a). On the convergence of bound optimization algorithms. In *Proc. 19th conference in uncertainty in artificial intelligence*.

Salakhutdinov, R., Roweis, S., & Ghahramani, Z. (2003b). Optimization with EM and expectation-conjugate-gradient. In *Proceedings of the twentieth international conference on machine learning*. Washington DC.

Salakhutdinov, R., Roweis, S., & Ghahramani, Z. (2004). *Relationship between gradient and EM steps in latent variable models*. University of Toronto. Retrieved July 19, 2012, from http://www.cs.toronto.edu/~rsalakhu/papers/report.pdf

Sandell, N. R., & Yared, K. I. (1978). *Maximum likelihood identification of state space models for linear dynamic systems* (tech. rep. No. ESL-R-814). Electronic

Systems Laboratory, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. Retrieved July 17, 2012, from http://hdl.handle.net/1721.1/1297

Särkkä, S. (2006). *Recursive bayesian inference on stochastic differential equations* (Doctoral dissertation, Helsinki University of Technology).

Särkkä, S. (2008). Unscented Rauch–Tung–Striebel smoother. *Automatic Control, IEEE Transactions on*, *53*(3), 845–849. doi:10.1109/TAC.2008.919531

Särkkä, S. (2012). Bayesian estimation of time-varying systems: discrete-time systems [Lectures notes]. Aalto University School of Science. Espoo. Retrieved August 20, 2012, from http://becs.aalto.fi/~ssarkka/course_k2012/full_course_booklet_2012.pdf

Särkkä, S., & Hartikainen, J. (2010). On Gaussian optimal smoothing of non-linear state space models. *Automatic Control, IEEE Transactions on*, *55*(8), 1938–1941. doi:10.1109/TAC.2010.2050017

Särkkä, S., & Sarmavuori, J. (2013, February). Gaussian filtering and smoothing for continuous-discrete dynamic systems. *Signal Processing*, *93*(2), 500–510. Advance online publication. doi:http://dx.doi.org/10.1016/j.sigpro.2012.09.002

Särkkä, S., Solin, A., Nummenmaa, A., Vehtari, A., Auranen, T., Vanni, S., & Lin, F.-H. (2012, January). Dynamic retrospective filtering of physiological noise in BOLD fMRI: DRIFTER. *NeuroImage*, *60*(2), 1517–1527. doi:10.1016/j.neuroimage.2012.01.067

Schön, T. B., Wills, A., & Ninness, B. (2011, January). System identification of nonlinear state-space models. *Automatica*, *47*(1), 39–49.

Segal, M., & Weinstein, E. (1989, May). A new method for evaluating the log-likelihood gradient, the Hessian, and the Fisher information matrix for linear dynamic systems. *Information Theory, IEEE Transactions on*, *35*(3), 682–687. doi:10.1109/18.30995

Shelley, K. H. (2007). Photoplethysmography: beyond the calculation of arterial oxygen saturation and heart rate. *Anesthesia & Analgesia*, *105*(6S Suppl), S31–S36.

Shumway, R. H., & Stoffer, D. S. (1982). An approach to time series smoothing and forecasting using the EM algorithm. *Journal of time series analysis*, *3*(4), 253–264.

The Mathworks Inc. (2012). Optimization toolbox user's guide. Natick, MA, USA. Retrieved July 25, 2012, from http://www.mathworks.se/help/toolbox/optim

Turner, R. E., & Sahani, M. (2011). Two problems with variational expectation maximization for time series models. In D. Barber, A. T. Cemgil & S. Chiappa (Eds.), *Bayesian time series models* (pp. 104–124). Cambridge University Press.

Wan, E. A., & Nelson, A. T. (2001). Dual extended Kalman filter methods. In S. Haykin (Ed.), *Kalman filtering and neural networks* (pp. 123–170). Wiley Online Library.

Wu, C. F. J. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics*, *11*(1), pp. 95–103.

Wu, Y., Hu, D., Wu, M., & Hu, X. (2006). A numerical-integration perspective on Gaussian filters. *Signal Processing, IEEE Transactions on*, *54*(8), 2910–2921. doi:10.1109/TSP.2006.875389