Parameter Estimation of Nonlinear Dynamic Systems

with Application to Failure Prognostics

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia, May 2011

Supervisor: Prof. Dr. Đani Juričić, Jožef Stefan Institute, Ljubljana, Slovenia

Evaluation Board:
Prof. Dr. Stanislav Strmčnik, Chairman, Jožef Stefan Institute, Ljubljana, Slovenia
Prof. Dr. Juš Kocijan, Member, Jožef Stefan Institute, Ljubljana, Slovenia
Dr. Sc. Miroslav Kárný, Member, Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Czech Republic

Matej Gašperin

Parameter Estimation of Nonlinear Dynamic Systems

with Application to Failure Prognostics

Doctoral Dissertation

Ocenjevanje parametrov nelinearnih dinamičnih sistemov

z uporabo za napovedovanje napak

Doktorska disertacija

Supervisor: Prof. Dr. Đani Juričić

May 2011

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL
Ljubljana, Slovenia

Contents

Abstract

This dissertation addresses the problem of parameter estimation of stochastic, discrete-time, state-space models from input and output data. This class of problems can be found in many modern engineering applications, where the relationships in the real world processes have to be inferred from the observed data and available prior knowledge. This problem is notoriously difficult because to estimate the model parameters one should dispose of full information on the internal system states. These are not directly observed and therefore also have to be inferred from the data. However, their estimation requires knowledge of the model parameters. This leads to the problem of joint estimation of both model parameters and hidden states.

In this dissertation, the framework for solving this problem is the expectation-maximization (EM) algorithm. It enables the computation of maximum likelihood estimates of the model parameters through recursive interplay between the parameter and the state estimation.

The central issue of the approach is related to the quality of estimation of internal states. Namely, state transitions in nonlinear systems modify the underlying state distributions so that a normally distributed initial state eventually renders into a distribution that can be far from normal. Apart from some special cases, no closed-form solutions are available. These distributions have to be suitably approximated in the expectation step of the EM algorithm, as they are essential for further maximization of the likelihood function. In the maximization step, the expected value of the likelihood function is maximized with respect to the model parameters. As the likelihood is a nonlinear function of random variables, computation of its expected value is again related to the problem of approximating the distributions of random variables.

The aim of this dissertation is to explore the performance of the EM algorithm under different approximation schemes, namely the sequential Monte Carlo (SMC) based approximations and the unscented transformation (UT). The SMC tends to reconstruct the entire probability density function, but it is limited by computational load, which increases dramatically with increasing number of states. This dissertation proposes a novel implementation of

the expectation-maximization algorithm, which we refer to as the UTEM algorithm, which is entirely based on the unscented transformation. The UT guarantees that the computational complexity is limited and the algorithm can thus be used for estimation of high dimensional models or applied to the problems, where computational load has to be low.

In this dissertation we provide a detailed performance analysis of the UTEM algorithm on simulated nonlinear dynamic systems. The focus of the estimation are the unknown model parameters, including the parameters of the system noise. We demonstrate that the proposed algorithm can solve the problem of joint estimation but has some limitations, dictated by the approximation errors. It seems that the convergence of the UTEM algorithm can be endangered mostly by divergence in noise covariance parameters originating in the insufficient quality of approximation. If the noise covariances are fixed, the estimation of the rest of the parameters converges much more reliably.

In the second part of the dissertation, the UTEM algorithm is applied to a relatively new problem domain dealing with estimation of the remaining useful life of mechanical systems. More particularly, the case study dealing with a gear transmission systems is addressed. The condition of the gear is assessed based on the vibrational signals. The relation between the damage progression in the gear and vibrational signature is described by a stochastic state-space model. Two model candidates are investigated: a simple linear black-box model and a more elaborated non-linear gray model. In both cases the unknown model parameters are estimated with the expectation-maximization algorithm. The resulting estimates are used to predict the future evolution of the damage and the remaining useful life (RUL) of the gearbox by applying Monte Carlo simulations. Surprisingly enough, no convincing advantage in the performance of the nonlinear model compared to the linear one has been observed. Actually, both of them provide reliable and useful estimates of the RUL. Hence timely alerts to the operators to plan maintenance actions can be issued well ahead of final failure. This feature represents a promising extension of functionality to the existing condition monitoring systems.

Povzetek

Doktorska disertacija obravnava problem ocenjevanja parametrov diskretnih stohastičnih modelov v prostoru stanj z uporabo vhodno-izhodnih podatkov. S potrebo po sintezi matematinega modela, ki opisuje relacije med podatki se pogosto srečujemo pri različnih inženirskih aplikacijah, kjer je potrebno zveze med razlinimi spremenljivkami v procesu doloiti le z uporabo merljivih količin ter predhodnega znanja o procesu. Dodaten izziv pri tem predstavlja dejstvo, da je za ocenjevanje parametrov modela v prostoru stanj potrebno poznati tudi vrednosti vseh nemerljivih stanj v procesu. Skrita stanja sicer lahko ocenimo iz podatkov, vendar postopki zahtevajo natančno poznavanje parametrov modela, kar nas pripelje do problema skupnega ocenjevanja parametrov ter nemerljivih stanj. Glavna tema te disertacije je iskanje uinkovitih reitev za ta tip problema.

V tem delu je kot okvir za reitev problema skupnega ocenjevanja parametrov ter stanj izbran postopek *Expectatiom-Maximization* (EM), ki omogoči izračun optimalne ocene parametrov po kriteriju funkcije verjetja preko iterativne optimizacije po neznanih parametrih in skritih stanjih modela.

Osrednji problem, s katerim se srečamo pri tem, je vprašanje kakovosti ocene skritih stanj v modelu. Ko imamo opravka z nelinearnimi relacijami, se namreč naključna spremenljivka z znano Gaussovo porazdelitveno funkcijo preslika v nakljuno spremenljivko s porazdelitveno funkcijo, ki je lahko precej drugana in jo je nemogoe izračunati analitično. Razen v redkih posebnih primerih tako postopkov ne moremo izraziti v zaprti obliki. Iskane porazdelitvene funkcije, ki jih potrebujemo v prvem koraku postopka EM, je potrebno izraziti v ustrezni poenostavljeni obliki oziroma kot približke. V drugem koraku postopka EM je cilj poiskati ekstrem pričakovane vrednosti funkcije verjetja v odvisnosti od parametrov modela. Ker je funkcija verjetja zopet nelinearna preslikava naključne spremenljivke, smo ponovno soočeni z problemom iskanja ustreznih priblikov porazdelitvene funkcije naključne spremenljivke.

V disertaciji raziskujemo obnaanje EM postopka pri uporabi različnih približkov za opis porazdelitvene funkcije naključnih spremenljivk. Osredotočili smo se na dve metodi: *sequential Monte Carlo* (SMC) in *unscented trans-*

form (UT). Ideja metode SMC je, da lahko porazdelitveno funkcijo opišemo poljubno natanno z uporabo velikega števila točk (t.i. delcev, ang. *particles*). Glavna omejitev pri uporabi te metode je njena računska zahtevnost, ki se drastično povečuje s povečevanjem dimenzije problema. Z namenom odprave te omejitve, disertacija predlaga novo izvedbo postopka EM (UTEM), ki v celoti temelji na uporabi računsko nezahtevne metode UT. S tem smo dosegli, da je računska zahtevnost postopka omejena in ga lahko uporabimo za ocenjevanje parametrov kompleksnih, visoko-dimenzionalnih modelov ali v primerih, ko je razpoložljiva ra˘unska mo omejena.

Novi postopek UTEM smo podrobno analizirali z uporabo simuliranih podatkov in s poudarkom na ocenjevanju parametrov nelinearnih modelov vključno s parametri naključnih spremenljivk. Rezultati pokaejo, da razviti postopek lahko uporabimo za reševanje takšnih problemov, vendar so se pri uporabi pokazale tudi njegove omejitve. Izkaže se, da ocenjevanje parametrov kovariančnih matrik šuma v modelu lahko še dodatno ogrozi konvergenco postopka. Vzrok za divergenco najdemo v prevelikih napakah pri iskanju približkov, ki so posledice postopka UT. Vendar, če vrednosti parametrov kovariance naključnih spremenljivk poznamo ali te niso predmet optimizacije, postopek teče in konvergira veliko bolj zanesljivo.

V drugem delu disertacije prikazujemo uporabo postopka UTEM za ocenjevanje preostale življenjske dobe mehanskih pogonov. Problematika predstavlja relativno novo smer razvoja metod za sprotno spremljanje in napovedovanje napak v mehanskih sistemih. V disertaciji smo se osredotočili na napovedovanje napak na zobnikih v reduktorju. O teh napakah lahko sklepamo na podlagi merjenih vibracij, ki so v primeru mehanskih sistemov enostavno merljive. V ta namen disertacija predlaga nov pristop, pri katerem povezavo med trenutnim stanjem zobnika in vibracijami na gredi modeliramo s stohastičnim modelom v prostoru stanj. Preizkusili smo dva različna modela; linearni model na podlagi črne škatlice ter podrobnejši nelinearni model. V obeh primerih parametre modela ocenjujemo sproti z uporabo postopka EM in iz trenutnih vrednosti značilke vibracij. Nato v vsakem koraku tako dobljeni model uporabimo za izračun napovedi gibanja poškobe v prihodnosti z uporabo Monte Carlo simulacije. Končni rezultat je ocena preostale življenjske dobe komponente.

Analiza rezultatov obeh modelov pokaže, da v primeru napovedovanja napak zobnikov uporaba nelinearnega modela ne izboljša kvalitete napovedi. Tako lahko točno in zanesljivo napoved dobimo tudi z uporabo bolj enostavnega lin-

earnega modela. To nam omogoča, da operaterju ali vzdrževalcu opreme lahko nudimo informacijo o trenutnem stanju napake in oceno časa do odpovedi, kar mu omogoči načrtovanje servisnih posegov v naprej ter s tem zniža stroške vzdrževanja in poveča dostopnost opreme. Nadgradnja z moduli za napovedovanje napak predstavlja naslednji korak v razvoju obstoječih sistemov za spremljanje stanja industrijske opreme.

Notation

General notation

$a, b, c, d, \alpha, \beta, \lambda, \kappa, t$	Scalars
$\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}, \mathbf{w}, \mathbf{v}$	Vectors
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{R}, \mathbf{Q}, \mathbf{X}, \mathbf{Y}$	Matrices

Symbols and operators

$\mathbf{A}$	State matrix
$\mathbf{B}$	Input matrix
$\mathbf{C}$	Measurement matrix
$\mathbf{D}$	Feedforward matrix
$E[\mathbf{x}]$	Expected value of the random variable $\mathbf{x}$
$E[f(\mathbf{x})]$	Expected value of the function of random variable $\mathbf{x}$
$L(\cdot)$	Log-likelihood function
$\mathbf{P}_0$	Initial state variance matrix
$\mathbf{P}_{t\|t}$	State covariance estimate at time t given information up to time t, i.e. filtered state covariance
$\mathbf{P}_{t\|T}$	State covariance estimate at time t given information up to time $T > t$, i.e. smoothed state covariance
$\mathbf{Q}$	Process noise covariance matrix
$\mathbf{R}$	Measurement noise covariance matrix
$\mathrm{Var}[\mathbf{x}]$	Variance of random variable $\mathbf{x}$
$\mathbf{X}_t$	System state vector values up to time t
$\mathbf{Y}_t$	System output values up to time t
α	Unscented transform scaling parameter
β	Unscented transform parameter

$\mathbf{f}(\cdot)$	System state function
$\mathbf{g}(\cdot)$	Measurement function
κ	Unscented transform parameter
$\mathcal{N}(\boldsymbol{\mu}, \mathbf{P})$	Gaussian distribution with mean value $\boldsymbol{\mu}$ and covariance matrix $\mathbf{P}$
N	State vector dimension
M	Number of particles
$p(\mathbf{x})$	Probability density function of $\mathbf{x}$
$\pi(\mathbf{x})$	Sampling probability density function
t	Time index
T	Time index
t^*	Time when the feature value reaches the critical value
$\mathbf{u}_t$	System input at time t
$\mathbf{x}_0$	Initial state value
$\bar{\mathbf{x}}$	Mean value of $\mathbf{x}$
$\dot{\mathbf{x}}(t)$	Time derivative of the state vector
$\mathbf{x}_t$	State vector at time t, $\mathbf{x}_t \in \mathbb{R}^N$
$\mathbf{x}_{1:t}$	State vector values from time 1 to t; $\mathbf{x}_t = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$
$\hat{\mathbf{x}}_{t\|t}$	State vector estimate at time t given information up to the time t, i.e. filtered estimate
$\hat{\mathbf{x}}_{t\|T}$	State vector estimate at time t given information up to the time $T > t$, i.e. smoothed estimate
$\mathcal{X}_i$	i-th sigma point representing the distribution $p(x)$
$\mathcal{Y}_i$	i-th sigma point representing the distribution $p(y)$
$\mathbf{y}_t$	Model output vector at time t
$\mathbf{y}_{1:t}$	Model output vector values from time 1 to t
$\mathbf{v}_t$	Measurement noise vector at time t
$\mathbf{w}_t$	Process noise vector at time t
$w_i^{(m)}$	i-th sigma point mean value weight
$w_i^{(c)}$	i-th sigma point covariance weight
$\gamma(\mathbf{x}^{(i)})$	Weight corresponding to particle $\mathbf{x}^{(i)}$
$\boldsymbol{\theta}$	Parameter vector
$\boldsymbol{\theta}_k$	Parameter vector estimate at iteration k
$\boldsymbol{\theta}^*$	Initial guess for the parameter vector
$\boldsymbol{\theta}'$	Fixed value of the parameter vector

$\widehat{\boldsymbol{\theta}}$	Parameter vector estimate
$\widehat{\boldsymbol{\theta}}_{ML}$	Maximum Likelihood estimate of parameter vector $\boldsymbol{\theta}$
$\widehat{\boldsymbol{\theta}}_{MAP}$	Maximum A Posteriori estimate of parameter vector $\boldsymbol{\theta}$
$\xrightarrow{\text{d}}$	Converges in distribution
$\xrightarrow{\text{p}}$	Converges in probability
$\xrightarrow{\text{a. s.}}$	Almost sure convergence
$\propto$	Proportional to

Abbreviations

i.i.d.	Independent and identically distributed
pdf	Probability density function
w.r.t	With respect to
CM	Condition monitoring
CMS	Condition monitoring system
FPT	First passage time
HMM	Hidden Markov model
EM	Expectation-maximization
EOL	End of life
KF	Kalman filter
KS	Kalman smoother
MC	Monte Carlo
MCMC	Markov chain Monte Carlo
ML	Maximum likelihood
MSE	Mean square error
PHM	Prognostics and Health Management
PF	Particle filter
PS	Particle smoother
RTS	Rauch-Tung-Striebel
RUL	Remaining useful life
SIR	Sequential importance resampling
SMC	Sequential Monte Carlo
SMCEM	Sequential Monte Carlo expectation-maximization
UKF	Unscented Kalman filter
UKS	Unscented Kalman smoother
UT	Unscented transformation
UTEM	Unscented transformation expectation-maximization

1 Introduction

Constructing mathematical models of the causal relationships in natural phenomena, is one of the core problems in science and technology. These models are parameterized by a set of unknown parameters and a way to compute their estimates is to use the available input-output data. Fundamentals for this reasoning process are traced back in the works of English priest and mathematician Reverend Thomas Bayes (Bayes, 1763) and German mathematician Carl Friedrich Gauss (Gauss, 1809). The birth of system identification as a separate discipline is marked by the work of Zadeh (1956). Since then, the field rapidly developed and resulted in a plethora of approaches and techniques. Nowadays, the field is considered mature and it offers engineering solutions supported by solid scientific theory (Ljung, 2008). The area is covered by numerous monographs, including Haykin (2001); Ljung (1999); Norton (2009); Shumway and Stoffer (2005); Söderström and Stoica (1989); Verhaegen and Verdult (2007). However, albeit the well-developed theory and many successful applications, the field still presents many challenges (Ninnes, 2009).

1.1 System identification problem

The term system identification refers to the process of estimating the unknown quantities in the model in a consistent and optimal way from noisy or incomplete data. The problem solving can be roughly divided into four major steps, which are usually executed sequentially (Ljung, 1999):

- **Design of experiments**. All system identification algorithms rely on measured data. The quality of the estimated model is directly related to the quality of the data. Design of experiment concerns the selection of the relevant independent process variables, that will be measured, as well as the design of their time profile (Goodwin and Payne, 1977).

- **Model class selection**. There are three major model classes, which can be distinguished as follows:

  - **First principle models**. The process of interest can be described by using equations based on the laws of physics. These models are usually referred to

as the *first principles models*.

- **Black-box models**. In black box models, variables and parameters usually do not have particular physical meaning as they are selected exclusively to mimic the behavior of the actual process.
- **Grey-box models**. It is always beneficial if any prior information about the system is included into the model. This strategy results in *grey box models*, where model structure can be defined in advance, but include parameters with unknown values.

- **Parameter estimation**. Based on the collected data and selected model structure, the missing values of model parameters have to be inferred. The optimal estimation procedure should include all available prior knowledge about the process into the parameter estimation procedure.

- **Model validation**. Finally, the model has to be validated before it can be used in any application. The validation should be performed with respect to the model purpose. For instance, model that will be used for implementation of advanced control algorithms may significantly differ from the model that will be used for model-based fault detection of the same process.

This dissertation deals only with the parameter estimation problem and it will be tacitly assumed that the model structure $\mathcal{M}$ fully coincides with the structure of the "true" process $\mathcal{M}^*$.

The model class considered in this dissertation is a *state-space model*. The state-space model consists of a set of equations describing the dynamic behavior of the system states and a set of equations describing the measurement process. All equations are parameterized by a set of fixed-value parameters. In mathematical terms, the model can be written as

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\theta}) + \mathbf{w}_t \tag{1.1a}$$

$$\mathbf{y}_t = \mathbf{g}(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\theta}) + \mathbf{v}_t \tag{1.1b}$$

where $\mathbf{x}_t$ is the internal state vector, $\mathbf{y}_t$ is the observed system output, $\mathbf{u}_t$ is the known system input, t is a discrete time index and $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are nonlinear functions parameterized by a parameter vector $\boldsymbol{\theta}$. Finally, $\mathbf{w}_t$ and $\mathbf{v}_t$ are stochastic processes describing process and measurement noise.

Although very practical for engineering applications, this model is demanding in terms of estimation. Depending on the nature of the unknown quantities, we can define three instances of the problem:

- estimation of the hidden system states ($\mathbf{x}_t$),
- estimation of unknown model parameters ($\boldsymbol{\theta}$),
- joint estimation of both states and parameters.

The system states are usually not directly observed and all information about them has to be inferred from noisy measurements. For that purpose, the knowledge of model parameters is needed. To estimate the model parameters, information about the system states variables is needed. Hence we get the problem of joint estimation of the model parameters and hidden system state variables, which is the central topic of this dissertation.

Different approaches to solve the joint estimation problem can be found in literature. The application of maximum likelihood for this problem is common and it is usually done by employing a gradient based search technique, such as a damped Gauss - Newton method to compute the estimates (Ljung, 1999; Söderström and Stoica, 1989). An alternative approach is to use the dual Kalman filtering framework (Haykin, 2001; Wan *et al.*, 2000; Yang *et al.*, 2010).

An interesting alternative was recently proposed by Gibson and Ninness (2005) and it is based on the expectation-maximization algorithm (Shumway and Stoffer, 2005). This algorithm originates from the statistics community and provides a procedure for computation of maximum likelihood estimates in the presence of missing data (Dempster *et al.*, 1977). Thus, if the hidden system states are considered as missing data, the expectation-maximization algorithm can be adopted for solving the joint estimation problem in state-space models. This algorithm was first applied to linear systems (Gibson and Ninness, 2005; Shumway and Stoffer, 2005). More recently authors are also considering its application to nonlinear systems using different methods to perform individual steps in the EM procedure (Chitralekha *et al.*, 2009; Schön *et al.*, 2011).

1.2 Purpose and contributions of the dissertation

The purpose of this dissertation is to present a specific approach to system identification of state-space models, where the expectation-maximization algorithm is used to compute the maximum likelihood estimates of the model parameters. For this purpose, the dissertation will provide a thorough overview of this approach and the already developed algorithms with the aim to identify their limitations. New algorithm for nonlinear system identification that has been developed with the aim to alleviate the problem of computational load is presented and compared with existing algorithms. Finally, the aim of this dissertation is to promote the implementation of novel identification methods in engineering applications.

The expectation-maximization algorithm was recently introduced to solve the problem of parameter estimation in the presence of unobserved system states (Gibson and Ninness, 2005; Shumway and Stoffer, 2005). Different implementations of the particular algorithm steps aimed at specific subclasses of models (Chitralekha *et al.*, 2009; Gibson and Ninness, 2005; Schön *et al.*, 2011) were presented. The purpose of this work is to perform a study of the existing methods and identify their limitations in terms of performance and computational load.

The novel algorithm that is presented in this dissertation is developed with the aim to decrease the computational load (compared to the algorithms based on the Monte-Carlo method) and improve the approximation accuracy (compared to the algorithms based on the extended Kalman filer). This is achieved using the unscented transformation for all steps of the EM algorithm.

In this dissertation we wish to contribute to an important yet emerging problem domain denoted as *failure prognostics* in rotational machinery. The goal of machine failure prognostics is to estimate the remaining useful life of the system. This is done by using the dynamic model of the degradation processes which has to be updated on-line by means of parameter estimation. Our solution of the problem of estimating the remaining useful life presents a contribution of this dissertation. This topic is continuously gaining importance in modern day industrial asset management. The solution below is novel and promising and has already gained a lot of attention from the relevant scientific community and industry.

1.2.1 Scientific contributions

The main original scientific contributions of this dissertation can be summarized as follows:

- **The UTEM algorithm for estimation of nonlinear state-space models using a computationally efficient approximation method**. In this dissertation, we present a novel algorithm for system identification of nonlinear systems (Gašperin and Juričić, 2011). The algorithm is based on the expectation-maximization algorithm and all the required approximations are done by unscented transformation. The new implementation alleviates the problem of computational load and guarantees that it is kept low even when the number of states increases.

- **Performance analysis of system identification algorithms in the framework of the expectation-maximization algorithm**. The performance of the UTEM algorithm is explored by means of simulated examples, which demonstrate its convergence properties. The UTEM algorithm is also put alongside the existing implementation of the EM algorithm, which is based on the sequential Monte

Carlo and the two algorithms are compared in terms of computational load and convergence properties.

- **Development of a novel approach to failure prognostics**. This dissertation proposes a model-based approach to machine failure prognostics. The idea is that the damage processes in mechanical drives can be modeled as a stochastic process. The dissertation presents two approaches for modeling the damage progression and the corresponding algorithms that can be used to estimate the model parameters on-line. The results were used for the estimation of the remaining useful life of an experimental gearbox (Gašperin *et al.*, 2009a,b,c, 2011a).

1.2.2 Publications

Publications by Matej Gašperin related to the dissertation topic:

- Gašperin, M., Boškoski, P., and Juričić, Đ. (2011). Model-based prognostics under non-stationary operating conditions. In *Proceedings of Annual Conference of the Prognostics and Health Management Society 2011, September 25. - 29. 2011, Montreal, Canada (Submitted)*.

- Gašperin, M., Juričić, Đ., Boškoski, P., and Vižintin, J. (2011). Model-based prognostics of gear health using stochastic nonlinear dynamic models. *International Journal of Condition Monitoring. (Submitted)*

- Gašperin, M., Juričić, Đ., Boškoski, P., and Vižintin, J. (2011). Model-based prognostics of gear health using stochastic dynamic models. *Mechanical Systems and Signal Processing*, **25**(2), 537-548.

- Gašperin, M. and Juričić, Đ. (2011). Application of unscented transformation in nonlinear system identification. In *18th IFAC World Congress, August 28. - September 2. 2011, Milano, Italy (Accepted for publication)*.

- Gašperin, M., Vrečko, D., and Juričić, Đ. (2010d). System identification of nonlinear dynamic models: Application to wastewater treatment plant. In *2010 IEEE Multiconference on System and Control : proceedings of the 19th International Conference on Control Applications, 10th IEEE International Symposium on Computer-Aided Control System Design (CACSD), 25th IEEE International Symposium on Intelligent Control (ISIC), September 8. - 10. 2010, Yokohama, Kanagawa, Japan*.

- Gašperin, M., Juričić, Đ., Boškoski, P., and Vižintin, J. (2010). Model-based prognostics of gear health using stochastic nonlinear dynamic models. In *The 7th In-*

ternational Conference on Condition Monitoring and Machinery Failure Prevention Technologies, June 22. - 24. 2010, Stratford-upon-Avon, England. .

- Gašperin, M., Juričić, Đ., and Boškoski, P. (2010). Condition prognosis of mechanical drives based on nonlinear dynamic models. In *Proceedings of Conference on Control and Fault-Tolerant Systems, SysTol'10, October 6. - 8. 2010, Nice, France* .

- Gašperin, M. and Juričić, Đ. (2010). Model-based prognostics of mechanical drives: the maximum-likelihood approach. In *Zbornik devetnajste mednarodne Elektrotehnike in raunalnike konference ERK 2010, September 20.- 22. 2010, Portorož, Slovenija.*

- Gašperin, M., Boškoski, P., and Juričić, Đ. (2009). Gear health monitoring and prognosis. In *Proceedings of the 10th International PhD Workshop on Systems and Control, September 22. - 26. 2009, Hluboka and Vitavou, Czech Republic.*

- Gašperin, M., Boškoski, P., and Juričić, Đ. (2009). Prognosis of gear health using stochastic dynamic models with online parameter estimation. In *Proceedings of Annual Conference of the Prognostics and Health Management Society 2009, September 27. - October 1. 2009, San Diego, USA.*

Other relevant results related to the dissertation topic:

- Gašperin, M., Boškoski, P., and Juričić, Đ., Svetek, A., Černe, S., Musizza B. (2010). Awarded with the second prize for innovation for the economy. In *3rd International Technology Transfer Conference, October 7. - 8. 2010, Ljubljana, Slovenia.*

Publications by Matej Gašperin not related to the dissertation topic:

- Gašperin, M. and Juričić, Đ. (2009). The uncertainty in burn prediction as a result of variable skin parameters: An experimental evaluation of burn-protective outfits. *Burns*, **35**(7), 970 – 982.

- Gašperin, M., Juričić, Đ., Musizza, B., and Mekjavič, I. (2008). A model-based approach to the evaluation of flame-protective garments. *ISA Transactions*, **47**(2), 198 – 210.

- Gašperin, M., Jovan, V., and Gradišar, D. (2008). Decision support system for polymerization production plant using PPIs. In *Proceedings of the 16th Mediterranean Conference on Control and Automation, June 25. - 27. 2008, Ajaccio, Corsica, France.*, pages 547 –551.

- Juričić, Đ., Musizza, B., Gašperin, M., Mekjavić, I., Vrhovec, M., and Dolanc, G. (2007). Evaluation of fire protective garments by using instrumented mannequin and model-based estimation of burn injuries. In *EUROSIM 2007 : proceedings of the 6th EUROSIM Congress on Modelling and Simulation, September 9. - 13. 2007, Ljubljana, Slovenia. Vol. 2, Full papers.*, 8 pages.

- Gašperin, M., Juričić, Đ., Musizza, B., Mekjavić, I., (2007). Estimation of burn injuries from temperature measurement used in evaluation of fire protective garments. In *Proceedings of the 15th Mediterranean Conference on Control and Automation, June 27. - 29. 2007, Athens, Greece.* , 6 pages.

Other results not related to the dissertation topic:

- Gašperin, M., Juričić, Đ., Musizza, B., and Mekjavič, I. (2009) 2009 ISA Transactions Best Paper Award.

1.3 Dissertation structure

The dissertation starts with an introduction of a mathematical model of dynamic system and its stochastic properties in Chapter 2. The state-space model will be successively derived from a more general stochastic differential-algebraic equation model. Depending on the conditions applied we come up with a nonlinear state-space models and a linear state-space model.

Chapter 3 provides algorithms for solving the state estimation problem for different subclasses of the models. First, Bayesian optimal filtering theory will be outlined, followed by the introduction of the Kalman filter/smoother for state estimation of linear models. Next, the algorithms for nonlinear state estimation will be derived, namely the unscented Kalman filter/smoother and the particle filter/smoother. The particular algorithms discussed here were selected because they represent an extremely useful set of tools for solving filtering problems: the Kalman filter for linear models, the unscented Kalman filter for high-dimensional problems where nonlinearities are not too "severe" and finally, the particle filter for highly nonlinear and non-Gaussian models.

Building on these results, Chapter 4 will develop the algorithms for joint state and parameter estimation in state-space models. The expectation-maximization algorithm will be presented. Particular steps will be solved with different approaches, depending on the class of the model. An important contribution is the introduction of the novel approximation-based algorithm for joint estimation in nonlinear models, which we refer to as the *unscented transformation expectation-maximization* (UTEM) algorithm.

The performance of the novel UTEM algorithm is studied and compared to the existing algorithms in Chapter 5. This is done using a simulated numerical example of a nonlinear time-invariant model. The important advantage of the novel UTEM algorithm lies in its low computational load, which becomes evident if the dimension of the state vector increases. The performance of the UTEM algorithm in this aspect is thus validated using a simulated four-dimensional state-space model of a wastewater treatment plant.

Chapter 6 deals with the application of the novel algorithm to the problem of estimation of remaining useful life of mechanical gearboxes. The dynamics of vibration signature is modeled by a state-space model. The progression of the damage process is described by hidden system states and the unknown parameters of the model need to be estimated on-line. Two different models will be applied and used to predict the future evolution of gear damage. This will enable the estimation of the remaining useful life of the gearbox.

The dissertation ends with conclusions and perspectives for future work.

2 Mathematical models of the dynamic systems

Different mathematical models that can reproduce the dynamical behavior of physical phenomena can be constructed. However, the estimation theory used in this dissertation focuses on a rather specific class of models, referred to as state-space models. The state-space models are represented by two sets of equations, one describing the evolution of system state variables in time and the other describing the measurement process. Since all natural phenomena also include stochastic components, this chapter demonstrates how the stochastic nature of the process is described in the state-space model formulation.

The most intuitive approach to the mathematical description of a real-world dynamic process would certainly be to construct a set of differential equations based on physical laws (e.g. conservation of mass, conservation of energy, laws of motion, etc.). They describe physical processes that govern the dynamic behavior of the entire system. The result of such an approach is a set of differential algebraic equations of the following form

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

where $\mathbf{F}(\cdot)$ is a function parameterized by a parameter vector $\boldsymbol{\theta}$, t denotes time, $\mathbf{x}(t)$ is a state variable, $\mathbf{u}(t)$ denotes the known external inputs and $\mathbf{w}(t)$ is a stochastic process, which stays for unmeasured system disturbances and unmodeled phenomena.

The measurement process is usually discrete, as the measurement instrumentation has finite sampling rate. Additionally, measurement equipment always introduces some measurement noise. The measurement process generally reads

$$\mathbf{G}(\mathbf{y}(t_k), \mathbf{x}(t_k), \mathbf{u}(t), \mathbf{v}(t_k), \boldsymbol{\theta}, t) = \mathbf{0} \tag{2.2}$$

where $\mathbf{G}(\cdot)$ is a function parameterized by $\boldsymbol{\theta}$, t_k is a discrete valued time, $\mathbf{y}(t_k)$ is the measured system output and $\mathbf{v}(t_k)$ is a stochastic process describing the measurement noise. Thus we get the stochastic differential-algebraic equation model.

Model 1: Stochastic differential-algebraic equation model

A general stochastic differential-algebraic equation model is given by

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

$$\mathbf{G}(\mathbf{y}(t_k), \mathbf{x}(t_k), \mathbf{u}(t), \mathbf{v}(t_k), \boldsymbol{\theta}, t) = \mathbf{0} \tag{2.3b}$$

where $\mathbf{w}(t)$ and $\mathbf{v}(t_k)$ are stochastic processes.

A more detailed definition and analysis of this class of systems can be found in Øksendal (2003).

Despite many activities and numerous attempts, an unified theory on how to handle the system identification of this general class of models still does not exist. The solutions have only been presented for certain special cases of Model 1, e.g. a deterministic case by Ascher and Petzold (1998); Brenan *et al.* (1996) or a stochastic case by Penski (2000); Schein and Denk (1998); Winkler (2004). One of the properties of stochastic differential-algebraic equation may be that it hides implicit differentiations of stochastic input $\mathbf{w}(t)$, which poses a serious problem if $\mathbf{w}(t)$ is a white noise.

A very important special subclass of Model 1 arises when (2.3) can be explicitly solved for $\dot{\mathbf{x}}(t)$,

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \boldsymbol{\theta}, t) \tag{2.4}$$

Instead of differential-algebraic equations, this model is governed by ordinary differential equations (ODE) and is commonly referred to as the *continuous-time state-space model*.

2.1 Towards discrete-time state-space models

Sampling of continuous-time systems results in data obtained at discrete points $\{1, 2, \ldots, t, t+1, \ldots\}$. Discrete point t corresponds to actual time $t \cdot T_0$, where T_0 is a sampling interval. Following the stochastic nature of the models, this section will lay down a probabilistic view and the basic model properties will be defined using probability theory. The probability density function of a random variable $\mathbf{x}_t$ will be denoted as $p(\mathbf{x}_t)$.

In order to simplify the derivations, we can suppress the existence of known input signals $\mathbf{u}(t)$ without the loss of generality. If the input signal is known, it can be shown that the state $\mathbf{x}_t$ is conditionally dependent on the input $\mathbf{u}_{t-1}$.

Property 2.1. (Markov property of the states)
The states $\{\mathbf{x}_t\}$ form a Markov sequence (or Markov chain if the state-space is discrete).

In other words, the discrete-time stochastic process $\{\mathbf{x}_t\}$ *is said to possess Markov property if*

$$p(\mathbf{x}_t|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{2.5}$$

and also

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_T) = p(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{2.6}$$

The Markov property means that the present $\mathbf{x}_t$ (and also the future) given $\mathbf{x}_{t-1}$ is independent of anything that has happened in the past (2.5) and furthermore, the past is independent of the future, given the present (2.6). Hence, the realization of the process at time t contains all the information about the past. This is sometimes also referred to as the *generalized causality principle*: the future can be predicted from knowledge of the present (Jazwinski, 1970).

Property 2.2. (Conditional independence of measurements)
The measurement $\mathbf{y}_t$ *given the state* $\mathbf{x}_t$ *is conditionally independent of the state history*

$$p(\mathbf{y}_t|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) = p(\mathbf{y}_t|\mathbf{x}_t) \tag{2.7}$$

Furthermore, the measurements are mutually conditionally independent over time

$$\begin{aligned} p(\mathbf{y}_t, \ldots, \mathbf{y}_T|\mathbf{x}_t, \ldots, \mathbf{x}_T) &= \prod_{i=t}^{T} p(\mathbf{y}_i|\mathbf{x}_t, \ldots, \mathbf{x}_T) \\ &= \prod_{i=t}^{T} p(\mathbf{y}_i|\mathbf{x}_i) \end{aligned} \tag{2.8}$$

Using the Markov property, the system model can be described as

$$\mathbf{x}_{t+1} \sim p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{x}_1, \ldots, \mathbf{x}_t) = p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{x}_t) \tag{2.9}$$

Where the notation $p_{\boldsymbol{\theta}}(\cdot)$ is used to describe a probability density function parameterized by parameter vector $\boldsymbol{\theta}$. The equation (2.9) describes the evolution of the system state over time. The initial state probability density function $p_{\boldsymbol{\theta}}(\mathbf{x}_1)$ is referred to as the *prior*. In general, the probability density function $p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{x}_t)$ can be non-Gaussian and can include nonlinearities.

The state process $\{\mathbf{x}_t\}$ is a Markov sequence, but it is usually not fully observable. The information about $\mathbf{x}_t$ is available indirectly through the measurement process based on the measurement model

$$\mathbf{y}_t|\mathbf{x}_t \sim p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t) \tag{2.10}$$

The above definitions describe a class of models referred to as the *hidden Markov model* (HMM) (Elliott *et al.*, 1995).

Model 2: Hidden Markov Model (HMM)

The HMM of a dynamic system is defined as

$$\mathbf{x}_{t+1}|\mathbf{x}_t \sim p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{x}_t), \tag{2.11a}$$

$$\mathbf{y}_t|\mathbf{x}_t \sim p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t) \tag{2.11b}$$

where the time recursions start from $p_{\boldsymbol{\theta}}(\mathbf{x}_1)$ and $\boldsymbol{\theta}$ denotes the time invariant parameter vector.

The class of hidden Markov models is still rather general. Applying certain limitations leads to a more specific cases, such as a discrete-time state-space model.

2.2 Discrete-time nonlinear state-space models

Let us assume that the relations for the system states evolution (2.11a) and measurements (2.11b) can be made explicit. That model can be written in the following form

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\theta}, \mathbf{w}_t, t) \tag{2.12a}$$

$$\mathbf{y}_t = \mathbf{g}(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\theta}, \mathbf{v}_t, t) \tag{2.12b}$$

Functions $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are nonlinear functions which describe the evolution of a state vector and measurement process over time. Additionally, $\mathbf{w}_t$ and $\mathbf{v}_t$ are white noise sequences. Introducing an additional restriction, that both process and measurement noise enter additively, leads to the most commonly used form referred to as the state-space model with additive noise (c.f. Model 3).

Model 3: Nonlinear state-space model with additive noise

Nonlinear state-space model with additive noise can be defined as

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\theta}, t) + \mathbf{w}_t \tag{2.13a}$$

$$\mathbf{y}_t = g(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\theta}, t) + \mathbf{v}_t \tag{2.13b}$$

where $\mathbf{w}_t$ and $\mathbf{v}_t$ are mutually independent white noise sequences and are uncorrelated with the input signal $\mathbf{u}_t$.

This model is actually a special case of the Model 2. This can be shown by using the following relations

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t, \boldsymbol{\theta}) = p_{\mathbf{w}}(\mathbf{x}_{t+1} - \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\theta}, t)) \tag{2.14a}$$

$$p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}) = p_{\mathbf{v}}(\mathbf{y}_t - \mathbf{g}(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\theta}, t)) \tag{2.14b}$$

where $p_{\mathbf{w}}(\cdot)$ and $p_{\mathbf{v}}(\cdot)$ correspond to the probability density functions of $\mathbf{w}_t$ and $\mathbf{v}_t$ (Jazwinski, 1970). The state-space model can also be illustrated as shown in Figure 2.1.

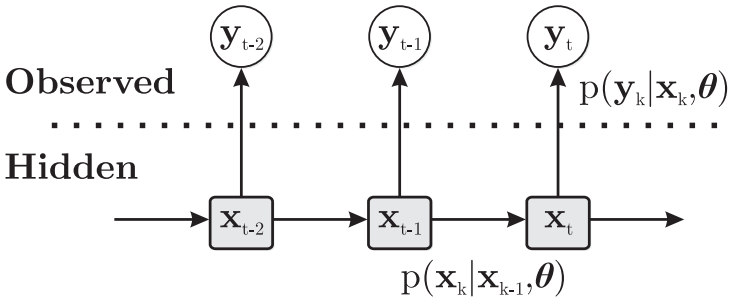


Figure 2.1: Graphical representation of a probabilistic state-space model.

Additional required condition is, that all the realizations of the random process are mutually independent, meaning that there is no information about the future realizations present in the past realizations. If this requirement is not met, the solution may not be traceable.

2.3 Discrete-time linear state-space models

The most simple version of Model 3 is the linear Gaussian state-space model. In that case $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ become linear functions and noise have Gaussian probability densities, denoted by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$.

Model 4: Linear state-space model

Linear state-space model with additive Gaussian noise is defined as

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t \tag{2.15a}$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t + \mathbf{v}_t \tag{2.15b}$$

where $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q})$ and $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R})$

In this case, the vector of model parameters $\boldsymbol{\theta}$ consist of elements of the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ and noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$.

$$\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{Q}, \mathbf{R}\} \tag{2.16}$$

The theory concerning linear state-space systems is well established and uncountable successful applications can be found in almost all areas of engineering.

2.4 Stochastic properties and noise distributions

2.4.1 Gaussian distribution

The Gaussian distribution is often used as a first approximation to describe real-valued random processes that cluster around a mean value.

Gaussian distribution

The probability density function of random vector $\mathbf{x}$ with mean value $\bar{\mathbf{x}}$ and covariance matrix $\mathbf{P_x}$ is given as

$$\mathcal{N}(\bar{\mathbf{x}}, \mathbf{P_x}) = \frac{1}{\sqrt{(2\pi)^{n_x} \det(\mathbf{P_x})}} e^{-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T \mathbf{P_x}^{-1}(\mathbf{x}-\bar{\mathbf{x}})} \tag{2.17}$$

where $n_x = \dim(\mathbf{x})$ and $\mathbf{P_x}$ is a positive definite covariance matrix with dimensions $n_x \times n_x$.

Lemma 2.1. Joint density of Gaussian variables (Gut, 2009)
Let two random variables $\mathbf{x}$ *and* $\mathbf{y}$ *have Gaussian probability density functions described by*

$$\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{P_x}) \tag{2.18a}$$

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{P_y}) \tag{2.18b}$$

then joint density of $\mathbf{x}, \mathbf{y}$ *and marginal distribution of* $\mathbf{y}$ *are given as*

$$\mathbf{x}, \mathbf{y} \sim \mathcal{N}\left(\begin{bmatrix} \bar{\mathbf{x}} \\ \mathbf{A}\bar{\mathbf{x}} + \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{P_x} & \mathbf{P_x}\mathbf{A}^T \\ \mathbf{A}\mathbf{P_x} & \mathbf{A}\mathbf{P_x}\mathbf{A}^T + \mathbf{P_y} \end{bmatrix} \right)$$

$$\mathbf{y} \sim \mathcal{N}(\mathbf{A}\bar{\mathbf{x}}, \mathbf{A}\mathbf{P_x}\mathbf{A}^T + \mathbf{P_y}) \tag{2.19a}$$

Lemma 2.2. Conditional density of Gaussian variables (Gut, 2009)
Let random variables $\mathbf{x}$ *and* $\mathbf{y}$ *have joint Gaussian probability density*

$$\mathbf{x}, \mathbf{y} \sim \mathcal{N}\left(\begin{bmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{y}} \end{bmatrix}, \begin{bmatrix} \mathbf{P_x} & \mathbf{P_{xy}} \\ \mathbf{P_{xy}}^T & \mathbf{P_y} \end{bmatrix} \right) \tag{2.20}$$

then marginal and conditional probability density functions of $\mathbf{x}$ *and* $\mathbf{y}$ *are Gaussian and given as*

$$\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{P_x}) \tag{2.21a}$$

$$\mathbf{y} \sim \mathcal{N}(\bar{\mathbf{y}}, \mathbf{P_y}) \tag{2.21b}$$

$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}(\bar{\mathbf{x}} + \mathbf{P_{xy}}\mathbf{P_y}^{-1}(\mathbf{y} - \bar{\mathbf{y}}), \mathbf{P_x} - \mathbf{P_{xy}}\mathbf{P_y}^{-1}\mathbf{P_{xy}}^T) \tag{2.21c}$$

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{y}} + \mathbf{P_{xy}}^T\mathbf{P_x}^{-1}(\mathbf{x} - \bar{\mathbf{x}}), \mathbf{P_y} - \mathbf{P_{xy}}^T\mathbf{P_x}^{-1}\mathbf{P_{xy}}) \tag{2.21d}$$

2.4.2 Transformations of Gaussian random variables

Linear transformation of Gaussian random variable

When a random variable with Gaussian probability density function $\mathbf{x}$ with mean value $\bar{\mathbf{x}}$ and covariance $\mathbf{P_x}$ undergoes a linear transformation

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b} \tag{2.22}$$

the result is a new random variable $\mathbf{y}$, which also has Gaussian distribution with the mean value $\bar{\mathbf{y}} = \mathbf{A}\bar{\mathbf{x}} + \mathbf{b}$ and covariance $\mathbf{P_y} = \mathbf{A}\mathbf{P_x}\mathbf{A}^T$.

Nonlinear transformation of Gaussian random variable

Assume that a random variable $\mathbf{x}$ with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P_x}$ undergoes a nonlinear transformation,

$$\begin{aligned} \mathbf{y} &= f(\mathbf{x}) \\ &= f(\bar{\mathbf{x}} + \boldsymbol{\xi}) \end{aligned} \tag{2.23}$$

where $\boldsymbol{\xi}$ is a zero mean random variable with covariance $\mathbf{P_x}$.

In general, the probability density function of random variable $\mathbf{y}$ can be of any form, depending on the nonlinear function $f(\cdot)$. To examine the posterior distribution more closely, let us look at the first two moments (mean and covariance) of the posterior distribution. For the purpose of this analysis, assume that the nonlinear transformation is analytic across the domain of $\mathbf{x}$ and infinitely many times differentiable, therefore it can be expressed as a multidimensional Taylor series. To calculate the expected value of the transformed random variable $\mathbf{y}$, we expand the function $f(\cdot)$ using the Taylor series around $\bar{\mathbf{x}}$.

$$\mathbf{y} = f(\bar{\mathbf{x}}) + D_{\boldsymbol{\xi}} f + \frac{1}{2!} D_{\boldsymbol{\xi}}^2 f + \frac{1}{3!} D_{\boldsymbol{\xi}}^3 f + \frac{1}{4!} D_{\boldsymbol{\xi}}^4 f + \dots \tag{2.24}$$

where $D_{\boldsymbol{\xi}}$ is an operator

$$D_{\boldsymbol{\xi}} = \sum_{j=1}^{n_x} \boldsymbol{\xi}_j \frac{\partial}{\partial \mathbf{x}_j} \tag{2.25}$$

which acts on $f(\cdot)$ on component-by-component basis. The i-th term in the Taylor series (2.24) is thus given by

$$\frac{1}{i!} D_{\boldsymbol{\xi}}^i f = \frac{1}{i!} \left[\sum_{j=1}^{n_x} \boldsymbol{\xi}_j \frac{\partial}{\partial \mathbf{x}_j} \right]^i f(\mathbf{x}) \Bigg|_{\mathbf{x}=\bar{\mathbf{x}}} \tag{2.26}$$

where n_x is the dimension of $\mathbf{x}$, $\boldsymbol{\xi}_j$ is the j-th component of $\boldsymbol{\xi}$ and $\mathbf{x}_j$ is the j-th component of $\mathbf{x}$. The expected value $\bar{\mathbf{y}}$ of $\mathbf{y}$ is

$$\begin{aligned}\bar{\mathbf{y}} &= E\left[f(\bar{\mathbf{x}}+\boldsymbol{\xi})\right] \\ &= f(\bar{\mathbf{x}}) + E\left[\frac{1}{2!}D^2_{\boldsymbol{\xi}}f + \frac{1}{3!}D^3_{\boldsymbol{\xi}}f + \frac{1}{4!}D^4_{\boldsymbol{\xi}}f + \dots\right]\end{aligned} \tag{2.27}$$

where the i-th term is given as

$$E\left[\frac{1}{i!}D^i_{\boldsymbol{\xi}}f\right] = \frac{1}{i!}E\left[\left(\sum_{j=1}^{n_x}\boldsymbol{\xi}_j\frac{\partial}{\partial\mathbf{x}_j}\right)^i f(\mathbf{x})\Big|_{\mathbf{x}=\bar{\mathbf{x}}}\right] \tag{2.28}$$

To calculate the mean value of $\mathbf{y}$ accurately to the m-th order of the Taylor series, all moments of $\mathbf{x}$ need to be known along with the derivatives of the function $f(\cdot)$ up to and including the m-th order term. For the use in subsequent analysis, let us write (2.27) in the following form

$$\bar{\mathbf{y}} = f(\bar{\mathbf{x}}) + \frac{1}{2}E\left[D^2_{\boldsymbol{\xi}}f\right] + E\left[\frac{1}{3!}D^3_{\boldsymbol{\xi}}f + \frac{1}{4!}D^4_{\boldsymbol{\xi}}f + \dots\right] \tag{2.29}$$

Noting that $E\left[\boldsymbol{\xi}_x\boldsymbol{\xi}_x^T\right] = \mathbf{P}_{\mathbf{x}}$, the second term can be written as

$$\begin{aligned}E\left[D^2_{\boldsymbol{\xi}}f\right] &= E\left[D_{\boldsymbol{\xi}}(D_{\boldsymbol{\xi}}f)^T\right] \\ &= E\left[(\nabla^T\boldsymbol{\xi}\boldsymbol{\xi}^T\nabla)f(\mathbf{x})\big|_{\mathbf{x}=\bar{\mathbf{x}}}\right] \\ &= (\nabla^T\mathbf{P}_{\mathbf{x}}\nabla)f(\mathbf{x})\big|_{\mathbf{x}=\bar{\mathbf{x}}}\end{aligned} \tag{2.30}$$

Rewriting (2.27) using notation from (2.30) yields

$$\bar{\mathbf{y}} = f(\bar{\mathbf{x}}) + \frac{1}{2}(\nabla^T\mathbf{P}_{\mathbf{x}}\nabla)f(\mathbf{x})\big|_{\mathbf{x}=\bar{\mathbf{x}}} + E\left[\frac{1}{3!}D^3_{\boldsymbol{\xi}}f + \frac{1}{4!}D^4_{\boldsymbol{\xi}}f + \dots\right] \tag{2.31}$$

Following the same procedure, we can derive expression for the covariance of the transformed random variable $\mathbf{y}$. By definition, the covariance $\mathbf{P}_{\mathbf{y}}$ is given by

$$\mathbf{P}_{\mathbf{y}} = E\left[(\mathbf{y}-\bar{\mathbf{y}})(\mathbf{y}-\bar{\mathbf{y}})^T\right] \tag{2.32}$$

where $\mathbf{y}-\bar{\mathbf{y}}$ can be obtained by substitutions from (2.27) and (2.24).

$$\begin{aligned}\mathbf{y}-\bar{\mathbf{y}} &= f(\bar{\mathbf{x}}+\boldsymbol{\xi}) - E\left[f(\bar{\mathbf{x}}+\boldsymbol{\xi})\right] \\ &= D_{\boldsymbol{\xi}}f + \frac{1}{2!}D^2_{\boldsymbol{\xi}}f + \frac{1}{3!}D^3_{\boldsymbol{\xi}}f + \frac{1}{4!}D^4_{\boldsymbol{\xi}}f + \dots \\ &\quad - E\left[\frac{1}{2!}D^2_{\boldsymbol{\xi}}f + \frac{1}{3!}D^3_{\boldsymbol{\xi}}f + \frac{1}{4!}D^4_{\boldsymbol{\xi}}f + \dots\right]\end{aligned} \tag{2.33}$$

Taking outer products and expectations, we can write covariance $\mathbf{P_y}$ as

$$\begin{aligned}\mathbf{P_y} = F_{\bar{\mathbf{x}}}\mathbf{P_x}F_{\bar{\mathbf{x}}}^T - \frac{1}{4}E\left[D_{\boldsymbol{\xi}}^2 f\right] E\left[D_{\boldsymbol{\xi}}^2 f\right]^T \\ + E\left[\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\frac{1}{i!j!}D_{\boldsymbol{\xi}}^i f\left(D_{\boldsymbol{\xi}}^j f\right)^T\right] \\ - \left(\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\frac{1}{2i!2j!}E\left[D_{\boldsymbol{\xi}}^{2i} f\right] E\left[D_{\boldsymbol{\xi}}^{2j} f\right]^T\right)\end{aligned} \tag{2.34}$$

where $F_{\bar{\mathbf{x}}} = \nabla_x f(\mathbf{x})\big|_{\mathbf{x}=\bar{\mathbf{x}}}$ is the Jacobian matrix of $f(\mathbf{x})$ evaluated at $\mathbf{x} = \bar{\mathbf{x}}$. The product of two infinite power series in (2.34) converges absolutely if the Taylor series of $f(\cdot)$ converges absolutely (Lang, 1999). In order to calculate the posterior covariance accurately, all the moments of random variable $\mathbf{x}$ and the derivatives of $f(\cdot)$ are required. The complete derivation of this result can be found in Julier and Uhlmann (1996).

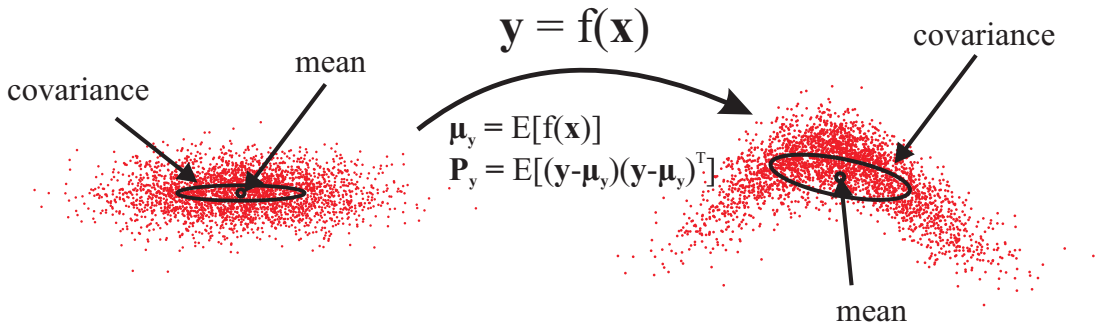


Figure 2.2: A nonlinear transformation is applied to a two-dimensional Gaussian random variable $\mathbf{x}$ (left) and the resulting random variable $\mathbf{y}$ is shown on the right, along with its mean and covariance.

The distribution of the transformed Gaussian random variable can have an infinite number of non-zero moments, which in general cannot be calculated analytically. We get a non-Gaussian random variable. The mean and covariance, shown in Figure 2.2 were calculated using $10,000$ samples drawn from prior distribution and then propagated trough the nonlinear function.

From the Figure 2.2 it can be seen, that in the case of non-Gaussian probability density functions the mean and covariance can have a limited descriptive power and additional moments may be required to form adequate approximations of the density function.

2.5 Chapter summary

This chapter reviewed the state-space dynamical models. We have highlighted one of the fundamental problems we come across when dealing with nonlinear stochastic models.

This is the calculation of probability density function of a random variable that undergoes a nonlinear transformation. We will return to this problem later on.

3 Optimal state estimation

Optimal state estimation relates to the problem of finding the estimates of the system states in a dynamical state-space model where the values of the model parameters are known in advance. Assume the set $\mathbf{Y}_T = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T\}$ represents all available measurements and we are interested in knowing $p(\mathbf{x}_t|\mathbf{Y}_T)$. Depending on the relation between t and T, the state estimation problem can take three different forms:

- $t > T$; the *prediction* problem
- $t = T$; the *filtering* problem
- $t < T$; the *smoothing* problem

We are reviewing this problem because in the context of the dissertation it will constitute an important ingredient of the system identification algorithms.

3.1 Bayesian state estimation

Consider the problem of estimating the probability density function of the state variable, given the measured system output in the system desribed by Model 2:

$$\mathbf{x}_{t+1}|\mathbf{x}_t \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t), \tag{3.1a}$$

$$\mathbf{y}_t|\mathbf{x}_t \sim p(\mathbf{y}_t|\mathbf{x}_t) \tag{3.1b}$$

Two operations on uncertainties are essential for further derivations. Assume three random variables $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$.

Basic operations of probability theory

Marginal distribution:

$$p(\mathbf{x}|\mathbf{z}) = \int p(\mathbf{x}, \mathbf{y}|\mathbf{z}) d\mathbf{y} \tag{3.2}$$

Joint distribution:

$$p(\mathbf{x}, \mathbf{y}|\mathbf{z}) = p(\mathbf{x}|\mathbf{y}, \mathbf{z}) p(\mathbf{y}|\mathbf{z}) \tag{3.3}$$

Solutions to all Bayesian estimation problems can be derived using the appropriate application of the above relations. The first one is the Bayes' theorem (Bayes, 1763). Using (3.2) and (3.3) we can write it as follows.

Bayes' theorem

Assuming stochastic variables $\mathbf{x}$,$\mathbf{y}$ and $\mathbf{z}$, Bayes' theorem is given by

$$p(\mathbf{x}|\mathbf{y},\mathbf{z}) = \frac{p(\mathbf{x},\mathbf{y}|\mathbf{z})}{p(\mathbf{y}|\mathbf{z})} = \frac{p(\mathbf{y}|\mathbf{x},\mathbf{z})p(\mathbf{x}|\mathbf{z})}{\int p(\mathbf{y}|\mathbf{x},\mathbf{z})p(\mathbf{x}|\mathbf{z})d\mathbf{x}} \tag{3.4}$$

The second standard formula is the *chain rule*. Consider the joint probability of T random variables $[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T]$ and assume a successive application of (3.3), only without the third variable $\mathbf{z}$. The result is the so-called chain rule (Durret, 1996).

Chain rule

Chain rule is given by

$$p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) = p(\mathbf{x}_1) \prod_{t=1}^{T} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \ldots, \mathbf{x}_1) \tag{3.5}$$

Using the Markov property and the chain rule, the joint distribution of the states $[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T]$ and measurements $[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T]$ are (c.f. (3.1))

$$p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) = p(\mathbf{x}_1) \prod_{t=1}^{T} p(\mathbf{x}_{t+1}|\mathbf{x}_t) \tag{3.6}$$

$$p(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) = \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t) \tag{3.7}$$

For a given T, the joint posterior distribution of the system states can be computed using Bayes' rule

$$p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T|\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T) = \frac{p(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)}{p(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T)} \tag{3.8}$$

As the numerator of the fraction in the right hand side of (3.8) is constant, the joint posterior distribution of the system states is proportional to the denominator.

$$p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T|\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T) \propto p(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) \tag{3.9}$$

Though appealing and straightforward, this kind of computation is not feasible, especially in real-time applications. The computational load increases every time a new measurement is obtained. This inconvenience can be alleviated by applying the concept of

transition probability density function. The basic result in relating the joint probability distributions is referred to as the Chapman-Kolmogorov equation (Papoulis, 1984).

For Markovian models it states that the transition probability of going from random variable $\mathbf{x}_1$ to another $\mathbf{x}_3$, i.e. $p(\mathbf{x}_3|\mathbf{x}_1)$ can be found from the transition probabilities of going form $\mathbf{x}_1$ to $\mathbf{x}_2$ and then from $\mathbf{x}_2$ to $\mathbf{x}_3$ by integrating over all possible values of $\mathbf{x}_2$.

Chapman-Kolmogorov equation

Let $m > n > p$ be integers that correspond to Markov sequence of random variables $\mathbf{x}_m, \mathbf{x}_n, \mathbf{x}_p$. The Chapman-Kolmogorov equation relates the transition densities as

$$p(\mathbf{x}_m|\mathbf{x}_p) = \int p(\mathbf{x}_m|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_p)d\mathbf{x}_n \tag{3.10}$$

Using Markov property (Chapter 2) and above relation, recursive algorithm for state estimation can be derived. This section will provide results to discrete-time optimal state estimation for filtering and smoothing problems.

3.1.1 Optimal discrete-time filtering

Bayesian filtering equations provide the basis for calculation of predicted distribution $p(\mathbf{x}_t|\mathbf{Y}_{t-1})$ and filtering distribution $p(\mathbf{x}_t|\mathbf{Y}_t)$. The joint distribution of $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ given $\mathbf{Y}_{t-1}$ can be written as

$$\begin{aligned} p(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{Y}_{t-1}) &= p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{Y}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1}) \\ &= p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{Y}_{t-1}) \end{aligned} \tag{3.11}$$

where we made use of the Markov property of the state. The marginal distribution over $\mathbf{x}_t$ can be obtained by integrating over $\mathbf{x}_{t-1}$.

$$p(\mathbf{x}_t|\mathbf{Y}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{Y}_{t-1})d\mathbf{x}_{t-1} \tag{3.12}$$

This gives the Chapman-Kolmogorov equation (3.10). Furthermore, the distribution of $\mathbf{x}_t$, given $\mathbf{Y}_t$ can be computed by application of the *Bayes' rule*,

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{Y}_t) &= \frac{1}{Z_t}p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{Y}_{t-1})p(\mathbf{x}_t|\mathbf{Y}_{t-1}) \\ &= \frac{1}{Z_t}p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_t) \end{aligned} \tag{3.13}$$

where the normalizing constant is given by

$$Z_t = \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_{t-1})d\mathbf{x}_t \tag{3.14}$$

Optimal discrete-time filtering (Simon, 2006)

1. Recursion starts from prior probability density function $p(\mathbf{x}_1)$.

2. The predictive distribution of the state $\mathbf{x}_k$ is computed using the *Chapman-Kolmogorov equation*.

$$p(\mathbf{x}_t|\mathbf{Y}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{Y}_{t-1})d\mathbf{x}_{t-1} \tag{3.15}$$

3. Given the measurement at time t, $\mathbf{y}_t$, the posterior distribution can be computed using the *Bayes' rule*.

$$p(\mathbf{x}_t|\mathbf{Y}_t) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_{t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_{t-1})d\mathbf{x}_t} \tag{3.16}$$

3.1.2 Optimal discrete-time smoothing

After obtaining the posterior distributions of the system state, the marginal posterior distributions conditional to all measurements up to time T can be calculated. Due to the Markov property of states, $\mathbf{x}_t$ is independent of $\{\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T\}$ given $\mathbf{x}_{t+1}$, which gives $p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_T) = p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_t)$. Using the Bayes' rule, we can write

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_T) &= p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_t) \\ &= \frac{p(\mathbf{x}_t, \mathbf{x}_{t+1}|\mathbf{Y}_t)}{p(\mathbf{x}_{t+1}|\mathbf{Y}_t)} \\ &= \frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{Y}_t)p(\mathbf{x}_t|\mathbf{Y}_t)}{p(\mathbf{x}_{t+1}|\mathbf{Y}_t)} \\ &= \frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_t)}{p(\mathbf{x}_{t+1}|\mathbf{Y}_t)} \end{aligned} \tag{3.17}$$

Using the above result, the joint distribution of $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ given $\mathbf{Y}_T$ can be computed as

$$\begin{aligned} p(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{Y}_T) &= p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_T)p(\mathbf{x}_{t+1}\mathbf{Y}_T) \\ &= p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_t)p(\mathbf{x}_{t+1}\mathbf{Y}_T) \\ &= \frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_t)p(\mathbf{x}_{t+1}\mathbf{Y}_T)}{p(\mathbf{x}_{t+1}|\mathbf{Y}_t)} \end{aligned} \tag{3.18}$$

where $p(\mathbf{x}_{t+1}\mathbf{Y}_T)$ is the smoothing distribution at time step $t+1$ and $p(\mathbf{x}_{t+1}|\mathbf{Y}_t)$ is given by

$$p(\mathbf{x}_{t+1}|\mathbf{Y}_t) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_t)d\mathbf{x}_t \tag{3.19}$$

The marginal distribution of $\mathbf{x}_t$ given $\mathbf{Y}_T$ is calculated by integrating over $\mathbf{x}_{t+1}$.

Optimal discrete-time smoothing (Simon, 2006)

1. Recursion starts from distribution $p(\mathbf{x}_T)$,
2. Compute the smoothed state estimate using the equation:

$$p(\mathbf{x}_t|\mathbf{Y}_T) = p(\mathbf{x}_t|\mathbf{Y}_t) \int \left[\frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_{t+1}|\mathbf{Y}_T)}{p(\mathbf{x}_{t+1}|\mathbf{Y}_t)} \right] d\mathbf{x}_{t+1} \tag{3.20}$$

where $p(\mathbf{x}_t|\mathbf{Y}_t)$ is the filtering distribution at time step t and $p(\mathbf{x}_{t+1}|\mathbf{Y}_t)$ is simply the predictive distribution of time step $t+1$.

3.2 Linear Kalman filter

The Bayesian framework enables the theoretical construction of optimal state estimates, but its computation may include the multi-dimensional integrals which, in general, do not have a closed-form solution. A special case are linear and Gaussian systems, for which there exists a closed-form solution, given by the *Kalman filter* (Kalman, 1960). This classical state estimation problem concerns linear system subjected to Gaussian noise.

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t \tag{3.21a}$$
$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t + \mathbf{v}_t \tag{3.21b}$$

where $\mathbf{y}_t$ is the observed output at time t, $\mathbf{x}_t$ is the system state at time t, $\mathbf{u}_t$ is the measured system input, $\mathbf{w}_t$ and $\mathbf{v}_t$ are a zero mean Gaussian i.i.d. vector stochastic processes with variance $\mathbf{Q}$ and $\mathbf{R}$, respectively. The prior distribution for $\mathbf{x}_1$ is also Gaussian $\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{P}_0)$. In probabilistic terms, the model is

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t+1}|\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t, \mathbf{Q}) \tag{3.22a}$$
$$p(\mathbf{y}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t|\mathbf{C}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t, \mathbf{R}) \tag{3.22b}$$

Our goal is to determine the probability density of $\mathbf{x}_t$, given all the measurement received up to time t, i.e., $\mathbf{Y}_t$.

3.2.1 Discrete-time linear Kalman filter

Due to the properties of Gaussian distribution and its linear transformation, the resulting distributions are also Gaussian. They can be evaluated in closed form (M.S. Grewal, 2008).

$$p(\mathbf{x}_{t+1}|\mathbf{Y}_t) = \mathcal{N}(\mathbf{x}_{t+1}|\hat{\mathbf{x}}_{t+1|t},\ \mathbf{P}_{t+1|t}) \tag{3.23}$$
$$p(\mathbf{x}_{t+1}|\mathbf{Y}_{t+1}) = \mathcal{N}(\mathbf{x}_{t+1}|\hat{\mathbf{x}}_{t+1|t+1},\ \mathbf{P}_{t+1|t+1}) \tag{3.24}$$
$$p(\mathbf{y}_{t+1}|\mathbf{Y}_t) = \mathcal{N}(\mathbf{y}_{t+1}|\mathbf{C}\hat{\mathbf{x}}_{t+1|t},\ \mathbf{C}\mathbf{P}_{t+1|t}\mathbf{C}^T + \mathbf{R}) \tag{3.25}$$

Algorithm 3.1: Linear Kalman filter

1. Initialize state estimate by $\hat{\mathbf{x}}_{0|0} = \bar{\mathbf{x}}_0$, $\mathbf{P}_{0|0} = \bar{\mathbf{P}}_0$ and set $t = 0$.

2. Prediction step:

$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{A}\hat{\mathbf{x}}_{t|t} + \mathbf{B}\mathbf{u}_t \tag{3.26}$$
$$\mathbf{P}_{t+1|t} = \mathbf{A}\mathbf{P}_{t|t}\mathbf{A}^T + \mathbf{Q} \tag{3.27}$$

3. Update step:

$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1|t}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{t+1|t}\mathbf{C}^T + \mathbf{R})^{-1} \tag{3.28}$$
$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + \mathbf{K}_{t+1}[\mathbf{y}_{t+1} - \mathbf{C}\hat{\mathbf{x}}_{t+1|t} - \mathbf{D}\mathbf{u}_{t+1}] \tag{3.29}$$
$$\mathbf{P}_{t+1|t+1} = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{C})\,\mathbf{P}_{t+1|t} \tag{3.30}$$

4. Set $t = t + 1$ and return to step 2.

3.2.2 Linear Kalman smoother

The discrete-time Kalman smoother, also called the discrete-time Rauch-Tung-Striebel (RTS) smoother (Bar-Shalom *et al.*, 2001; Rauch *et al.*, 1965), provides a closed form solution to optimal smoothing problem, which is the distribution

$$p(\mathbf{x}_t|\mathbf{Y}_T) = \mathcal{N}(\mathbf{x}_t|\hat{\mathbf{x}}_{t|T}, \mathbf{P}_{t|T}) \tag{3.31}$$

The difference to the previous result is that the solution is conditional on all measurement data $\mathbf{Y}_T$ and not on only the first t steps.

Algorithm 3.2: Kalman Smoother

1. Run the Kalman filter algorithm and store the state estimates and covariances.
2. Initialize the algorithm with $\hat{\mathbf{x}}_{T|T}$, $\mathbf{P}_{T|T}$ and set $t = T$.
3. Compute smoothed state estimates by

$$\mathbf{S}_t = \mathbf{P}_{t|t}\mathbf{A}^T\mathbf{P}_{t+1|t}^{-1} \tag{3.32}$$

$$\hat{\mathbf{x}}_{t|T} = \hat{\mathbf{x}}_{t|t} + \mathbf{S}_t(\hat{\mathbf{x}}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) \tag{3.33}$$

$$\mathbf{P}_{t|T} = \mathbf{P}_{t|T} + \mathbf{S}_t(\mathbf{P}_{t+1|T} - \mathbf{P}_{t+1|t})\mathbf{S}_t^T \tag{3.34}$$

4. Set $t = t - 1$ and go back to step 3.

3.2.3 Lag-one covariance smoother

In the next chapter we will also require a result from a modified smoother, usually referred to as the lag-one covariance smoother (Shumway and Stoffer, 2005). With $\mathbf{K}_t$, $\mathbf{S}_t$ and $\mathbf{P}_{t|T}$ obtained from filtering and smoothing equations and using the initial condition

$$\mathbf{M}_{T-1|T} = (\mathbf{I} - \mathbf{K}_T\mathbf{C})\mathbf{A}\mathbf{P}_{T-1|T-1} \tag{3.35}$$

where T is the number of measurements available. The lag-one covariance can be computed by reverse time recursion of the following expression

$$\mathbf{M}_{t|T} = \mathbf{P}_{t|t}\mathbf{S}_{t-1}^T + \mathbf{S}_t(\mathbf{M}_{t+1|T} - \mathbf{A}\mathbf{P}_{t|t})\mathbf{S}_{t-1}^T \tag{3.36}$$

Example 3.1. Linear Kalman filter for Gaussian random walk
One-dimensional Gaussian random walk model is given by

$$x_{t+1} = x_t + w_t, \qquad w_t \sim \mathcal{N}(0, q) \tag{3.37a}$$

$$y_t = x_t + v_t, \qquad v_t \sim \mathcal{N}(0, r) \tag{3.37b}$$

Assuming that we observe the measurements y_t and wish to estimate the values of system states x_t. In this example, the noise covariances are scalar and their values are $q = 1$ and $r = 2$. The Kalman filter state estimates are shown in Figure 3.1 and the Kalman smoother estimates in Figure 3.2

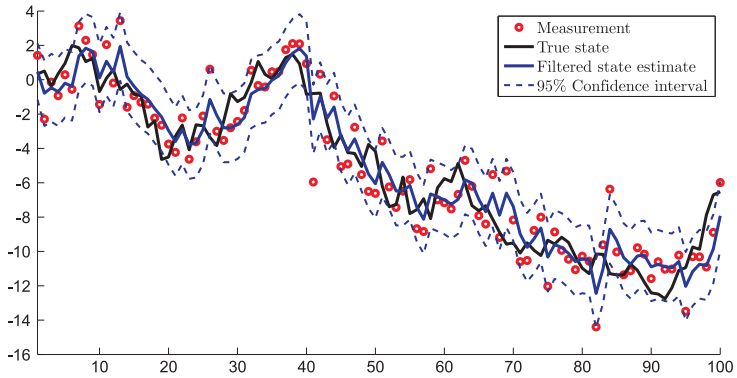


Figure 3.1: Measurements, true state and the Kalman filter estimates of Example 3.1.

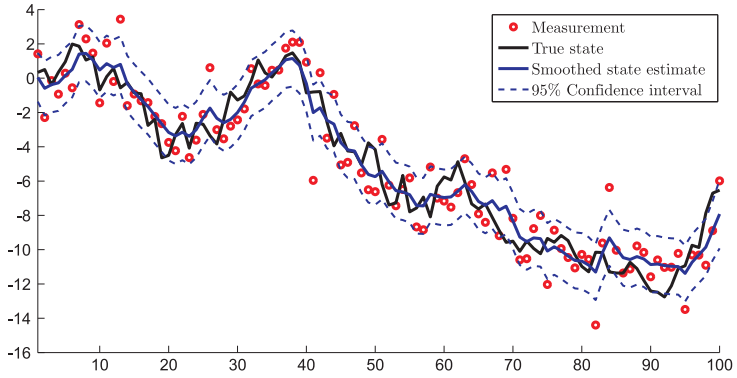


Figure 3.2: Measurements, true state and the Kalman smoother estimates of Example 3.1.

3.3 Sigma point Kalman filtering

In optimal filtering of nonlinear systems we encounter a problem related to the nonlinear transformation of random variables as indicated in Chapter 2. The nonlinear transformation of a Gaussian random variable can result in a probability distribution that cannot be described by a finite number of moments. Therefore, the exploitation of nice properties of Gaussian random variables is no longer possible and one is forced to incorporate approximations into the estimation procedure.

The most straightforward idea is to compute a local linear approximation of the nonlinear function and proceed in the same way as in the Kalman filter. The only difference would be that linear system matrices are replaced by the Jacobians of the nonlinear functions, evaluated at the current state estimate. This algorithm is called the *Extended Kalman filter* (EKF) (Bar-Shalom *et al.*, 2001; Jazwinski, 1970; Maybeck, 1982). It has long been the only applied solution of nonlinear filtering problem and is still frequently used today. A nice property of the EKF is that it retains all the benefits of the Kalman filter in terms of low computational complexity and transparency. A drawback is that

it requires computation of the derivatives, which may sometimes not be straightforward. Additionally, the accuracy of this approximation can be relatively poor which often leads to divergence of the EKF.

Partial remedy of this problem was developed by Julier and Uhlmann (1996). Their approach is based on the argument that it is easier to approximate probability distributions rather than arbitrary nonlinear functions. Their method is referred to as the unscented transformation and has proved to be more efficient than the solution based on the linearizion (e.g. the EKF). This section will present the basic unscented transformation method and some of its extensions. Based on this, the unscented Kalman filter and smoother will be developed. Some additional attention will be paid to the analysis of the approximation error introduced by the approximations.

3.3.1 Unscented transformation

The unscented transformation (UT) (Julier and Uhlmann, 1996, 2004) can be used to form approximations to the distribution of random variable that undergoes a nonlinear transformation. The idea mostly builds on the assumption that it is easier to approximate a probability density function than an arbitrary nonlinear function. Considering $\mathbf{x}$ to be an N dimensional random variable with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{\mathbf{x}}$, that is propagated trough a nonlinear function $\mathbf{f}(\cdot)$. The result is a new random variable $\mathbf{y}$.

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \tag{3.38}$$

Let $\mathbf{x}$ have an arbitrary probability density function $p(\mathbf{x})$. The aim is to approximate the probability density function of $\mathbf{y}$ by selecting a set of weighted samples, also called sigma points. The set of sigma points and their corresponding weights $\{\boldsymbol{\mathcal{X}}_i, w_i\}$ is such that it correctly captures the first two moments (mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{\mathbf{x}}$) of the $N-$dimensional random variable $\mathbf{x}$. The selection scheme, first proposed by Julier and Uhlmann (1996), that satisfies this requirement is as follows (Särkkä, 2008; van der Merwe, 2004):

$$\boldsymbol{\mathcal{X}}_0 = \bar{\mathbf{x}} \qquad i = 0 \tag{3.39a}$$

$$\boldsymbol{\mathcal{X}}_i = \bar{\mathbf{x}} + \left(\sqrt{(N+\kappa)\mathbf{P}_x}\right)_i \qquad i = 1, \ldots, N \tag{3.39b}$$

$$\boldsymbol{\mathcal{X}}_i = \bar{\mathbf{x}} - \left(\sqrt{(N+\kappa)\mathbf{P}_x}\right)_i \qquad i = N+1, \ldots, 2N \tag{3.39c}$$

and the corresponding weights as

$$w_i = \frac{\kappa}{N+\kappa} \qquad i = 0 \tag{3.40a}$$

$$w_i = \frac{1}{2\,(N+\kappa)} \qquad i = 1, \ldots, 2N \tag{3.40b}$$

where w_i is the weight associated with i-th sigma point. The weights satisfy the condition $\sum_{i=0}^{2n} w_i = 1$. Additionally, $\left(\sqrt{(N+\kappa)\mathbf{P}_x}\right)_i$ is the i-th column of the matrix square root of the weighted covariance matrix and κ is a scaling parameter. It is important to note that a matrix square root of a positive-definite matrix is not unique. Therefore, any orthonormal rotation of the sigma-point set is again a valid set.

Each sigma-point $\boldsymbol{\mathcal{X}}_i$ is then propagated trough a nonlinear function

$$\boldsymbol{\mathcal{Y}}_i = \mathbf{f}(\boldsymbol{\mathcal{X}}_i) \qquad i = 0, \ldots 2N \tag{3.41}$$

and the approximated mean, covariance $\mathbf{P}_y$ and cross covariance $\mathbf{P}_{xy}$ are obtained as follows:

$$\bar{\mathbf{y}} \approx \sum_{i=0}^{2N} w_i \boldsymbol{\mathcal{Y}}_i \tag{3.42}$$

$$\mathbf{P}_y \approx \sum_{i=0}^{2N} w_i (\boldsymbol{\mathcal{Y}}_i - \bar{\mathbf{y}})(\boldsymbol{\mathcal{Y}}_i - \bar{\mathbf{y}})^T \tag{3.43}$$

$$\mathbf{P}_{xy} \approx \sum_{i=0}^{2N} w_i (\boldsymbol{\mathcal{X}}_i - \bar{\mathbf{x}})(\boldsymbol{\mathcal{Y}}_i - \bar{\mathbf{y}})^T \tag{3.44}$$

The posterior mean value and covariance obtained using this approximation are accurate to the second order of the Taylor series expansion of $\mathbf{f}(\mathbf{x})$ for *any* nonlinear function (Julier and Uhlmann, 1996, 2004; van der Merwe, 2004). Errors in higher orders can be influenced by the selection of the transformation parameter κ, depending on the probability density function and the type of nonlinear functions $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$.

The proposed selection scheme (3.39) and (3.40) has a property that as the dimension N increases the sigma points move away from the mean value. While this does not affect the correct capturing of the prior mean and covariance, it can lead to sampling possible non-local effects in the case of multi-modal density functions. To address this issue, the transformation parameter κ can be used to scale the sigma points towards or away from the mean of the prior distribution. By examining (3.39) one can notice, that the points are scaled further from $\bar{\mathbf{x}}$ when $\kappa > 0$ and closer to it when $\kappa < 0$. In case $\kappa = 0$, the points scale away from the mean proportional to $\sqrt{N}$. The dimensional scaling invariance is achieved for the special case $\kappa = 3 - N$ by canceling the effect of N. However, when $\kappa = 3 - N < 0$, the weight $w_0 < 0$ and the approximated posterior covariance is no longer positive-semidefinite. To alleviate this problem, the scaled unscented transformation was proposed (Julier, 2002).

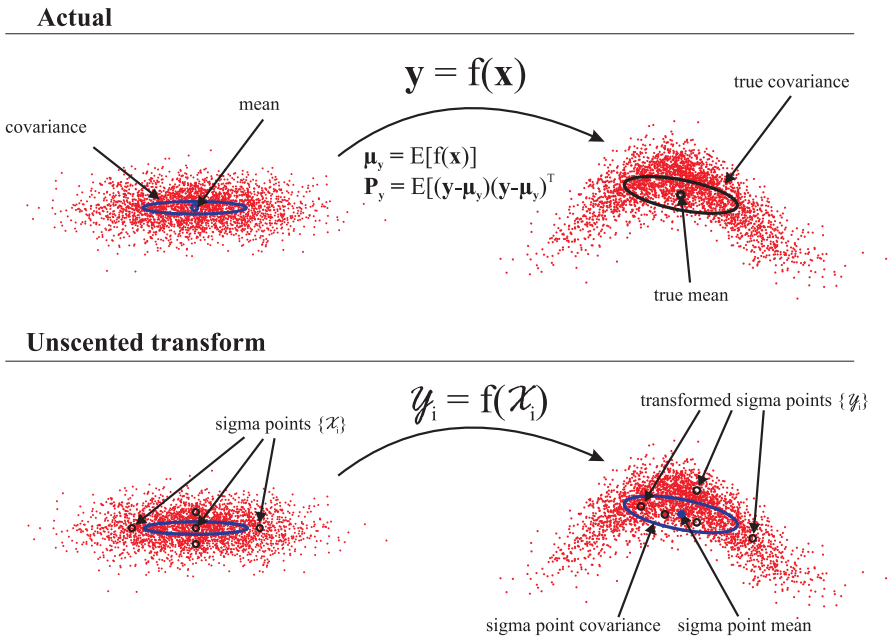


Figure 3.3: Demonstration of posterior mean and covariance approximation using unscented transformation.

Scaled unscented transformation

The main idea of a scaled unscented transformation is to replace the original set of sigma-points with a transformed set given by:

$$\boldsymbol{\mathcal{X}}'_i = \boldsymbol{\mathcal{X}}_0 + \alpha(\boldsymbol{\mathcal{X}}_i - \boldsymbol{\mathcal{X}}_0) \quad i = 1, \ldots 2N \tag{3.45}$$

where α is a scaling parameter ($0 < \alpha < 1$). This scaling parameter gives an extra degree of freedom to the sigma-point scheme, without causing the resulting covariance to possibly become non-positive semidefinite. To reduce the number of calculations, the scaling of sigma-points can be incorporated into the selection scheme, which is then given as:

$$\boldsymbol{\mathcal{X}}'_i = \bar{\mathbf{x}}; \qquad i = 0 \tag{3.46a}$$

$$\boldsymbol{\mathcal{X}}'_i = \bar{\mathbf{x}} + \left(\sqrt{(N+\lambda)\mathbf{P}_{\mathbf{x}}}\right)_i; \qquad i = 1, \ldots, N \tag{3.46b}$$

$$\boldsymbol{\mathcal{X}}'_i = \bar{\mathbf{x}} - \left(\sqrt{(N+\lambda)\mathbf{P}_{\mathbf{x}}}\right)_i; \qquad i = N+1, \ldots, 2N \tag{3.46c}$$

and the corresponding weights:

$$w_i^{(m)} = \frac{\lambda}{N+\lambda}; \qquad i = 0 \tag{3.47a}$$

$$w_i^{(c)} = \frac{\lambda}{N+\lambda} + (1 - \alpha^2 + \beta); \qquad i = 0 \tag{3.47b}$$

$$w_i^{(m)} = w_i^{(c)} = \frac{1}{2(N+\lambda)}; \qquad i = 1, \ldots, 2N \tag{3.47c}$$

where $w_i^{(m)}$ and $w_i^{(c)}$ are the weights associated with i-th sigma point used in approximation of the mean value and the covariance, respectively. Additionally, they satisfy the conditions $\sum_{i=0}^{2N} w_i^{(m)} = 1$ and $\sum_{i=0}^{2N} w_i^{(c)} = 1$. Furthermore, λ is the scaling parameter defined by (Julier, 2002):

$$\lambda = \alpha^2 (N + \kappa) - N \tag{3.48}$$

The presented form of the unscented transformation includes three parameters (κ,α,β) that control the placement of the *sigma points* around the mean value. Parameter β affects the weighting of the zeroth sigma point for the calculation of the covariance and can be used to incorporate prior knowledge about the distribution. For a Gaussian prior the optimal value is $\beta = 2$ (Julier, 2002). Choosing $\kappa \geq 0$ ensures positive semidefinitness of the covariance matrix although it is still an open question weather there exists a global optimal selection. Similar holds true for α that should lie in the interval $0 \leq \alpha \leq 1$. Smaller values of α generally help to avoid sampling non-local effects in the case when nonlinearities are strong.

The approximated posterior mean $\bar{y}$ can be calculated using the scaled unscented transform as

$$\boldsymbol{\mathcal{Y}}_i' = \mathbf{f}(\boldsymbol{\mathcal{X}}_i') \tag{3.49}$$

$$\bar{\mathbf{y}} \approx \sum_{i=0}^{2N} w_i^{(m)} \boldsymbol{\mathcal{Y}}_i' \tag{3.50}$$

$$\mathbf{P}_{\mathbf{y}} \approx \sum_{i=0}^{2N} w_i^{(c)} (\boldsymbol{\mathcal{Y}}_i' - \bar{\mathbf{y}})(\boldsymbol{\mathcal{Y}}_i' - \bar{\mathbf{y}})' \tag{3.51}$$

$$\mathbf{P}_{\mathbf{xy}} \approx \sum_{i=0}^{2N} w_i^{(c)} (\boldsymbol{\mathcal{X}}_i' - \bar{\mathbf{x}})(\boldsymbol{\mathcal{Y}}_i' - \bar{\mathbf{y}})^T \tag{3.52}$$

The three transformation parameters (κ, α, β) affect the errors in the higher order terms (Julier and Uhlmann, 2004). However, a global optimal setting of the parameters has not yet been determined. Empirical evidence suggest, that the optimal setting is case dependent and furthermore, that the exact values of the parameters are not critical as long as they result in a numerically well-behaved set of sigma-points and the corresponding weights (van der Merwe, 2004; Wan and van der Merwe, 2000).

Scaled unscented transformation

1. Select the sigma-points and their weights according to (3.46) and (3.47).

$$\{\boldsymbol{\mathcal{X}}_i, w_i^{(m)}, w_i^{(c)}\}; \qquad i = 1, \ldots, 2N \tag{3.53}$$

2. Propagate the sigma-points through the function $\mathbf{f}(\cdot)$.

$$\mathcal{Y}_i = \mathbf{f}(\mathcal{X}_i) \qquad i = 1, \ldots, 2N \tag{3.54}$$

3. Compute the estimated posterior mean and covariance according to (3.50) and (3.51).

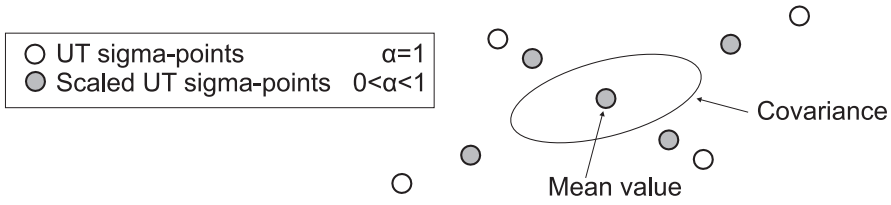


Figure 3.4: Example of sigma-points placement for unscented transformation and scaled unscented transformation.

Approximated posterior distribution

We can examine the approximation errors introduced through the unscented transformation by analyzing the posterior distribution statistics using the Taylor series expansion. Assume an N dimensional random variable $\mathbf{x}$ with mean value $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_\mathbf{x}$ that undergoes a nonlinear transformation $\mathbf{f}(\cdot)$, which is infinitely many times differentiable and can therefore be expressed by a Taylor series.

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \mathbf{f}(\bar{\mathbf{x}} + \boldsymbol{\sigma}_\mathbf{x}) \tag{3.55}$$

where $\boldsymbol{\sigma}_\mathbf{x}$ is a zero mean random variable with covariance $\mathbf{P}_\mathbf{x}$. The sigma-points are given as

$$\begin{aligned} \mathcal{X}_i &= \bar{\mathbf{x}} \pm \left(\sqrt{(N+\lambda)\mathbf{P}_x}\right)_i \\ &= \bar{\mathbf{x}} \pm \sqrt{(N+\lambda)}\boldsymbol{\xi}_i \\ &= \bar{\mathbf{x}} \pm \tilde{\boldsymbol{\xi}}_i \end{aligned} \tag{3.56}$$

where $\tilde{\boldsymbol{\xi}}_i = \left(\sqrt{(N+\lambda)\mathbf{P}_x}\right)_i$ is the i-th column of matrix square root of the weighted covariance matrix $\mathbf{P}_x$, $\boldsymbol{\xi}_i$ is the i-th column of the matrix square root of $\mathbf{P}_x$, and $\bar{\mathbf{x}}$ is the mean value of $\mathbf{x}$. Additionally, the following relation holds

$$\sum_{i=1}^{N} \boldsymbol{\xi}_i \boldsymbol{\xi}_i^T = \mathbf{P}_\mathbf{x} \tag{3.57}$$

Given the formulation of sigma points we can also write the Taylor series expansion for each point propagation through nonlinear function $\mathbf{f}(\cdot)$.

$$\boldsymbol{\mathcal{Y}}_i = \mathbf{f}(\boldsymbol{\mathcal{X}}_i) = \mathbf{f}(\bar{\mathbf{x}}) + D_{\tilde{\boldsymbol{\xi}}_i}\mathbf{f} + \frac{1}{2!}D^2_{\tilde{\boldsymbol{\xi}}_i}\mathbf{f} + \frac{1}{3!}D^3_{\tilde{\boldsymbol{\xi}}_i}\mathbf{f} + \frac{1}{4!}D^4_{\tilde{\boldsymbol{\xi}}_i}\mathbf{f} + \dots \tag{3.58}$$

where $D_{\boldsymbol{\xi}_i}\mathbf{f}$ is defined as in (2.25), repeated here for convenience. The i-th term is thus given by

$$\frac{1}{i!}D^i_{\tilde{\boldsymbol{\xi}}}\mathbf{f} = \frac{1}{i!}\left[\sum_{j=1}^{N}\tilde{\boldsymbol{\xi}}_j\frac{\partial}{\partial \mathbf{x}_j}\right]^i \mathbf{f}(\mathbf{x})\Big|_{\mathbf{x}=\bar{\mathbf{x}}} \tag{3.59}$$

Using the unscented transformation, the posterior mean is given by

$$\begin{aligned}\bar{\mathbf{y}} &= \frac{\lambda}{N+\lambda}\mathbf{f}(\bar{\mathbf{x}}) + \frac{1}{2(N+\lambda)}\sum_{i=1}^{2N}\left[\mathbf{f}(\bar{\mathbf{x}}) + \frac{1}{2!}D^2_{\tilde{\boldsymbol{\xi}}_i}\mathbf{f} + \frac{1}{3!}D^3_{\tilde{\boldsymbol{\xi}}_i}\mathbf{f} + \dots\right] \\ &= \mathbf{f}(\bar{\mathbf{x}}) + \frac{1}{2(N+\lambda)}\sum_{i=1}^{2N}\left[\frac{1}{2!}D^2_{\tilde{\boldsymbol{\xi}}_i}\mathbf{f} + \frac{1}{4!}D^4_{\tilde{\boldsymbol{\xi}}_i}\mathbf{f} + \dots\right]\end{aligned} \tag{3.60}$$

where the fact that the sigma points are distributed symmetrically around $\bar{\mathbf{x}}$ results in zero odd moment terms. The second term in (3.60) can be rewritten as

$$\begin{aligned}\frac{1}{2(N+\lambda)}\sum_{i=1}^{2N}\frac{1}{2}D^2_{\tilde{\boldsymbol{\xi}}_i}\mathbf{f} &= \frac{1}{2(N+\lambda)}\sum_{i=1}^{2N}\frac{1}{2}\left[D_{\tilde{\boldsymbol{\xi}}_i}(D_{\tilde{\boldsymbol{\xi}}_i}\mathbf{f})^T\right] \\ &= \frac{1}{4(N+\lambda)}\sum_{i=1}^{2N}\left[(\nabla^T\tilde{\boldsymbol{\xi}}_i\tilde{\boldsymbol{\xi}}_i^T\nabla)\mathbf{f}(\mathbf{x})\Big|_{\mathbf{x}=\bar{\mathbf{x}}}\right] \\ &= \frac{1}{4(N+\lambda)}\sum_{i=1}^{2N}\left[(\nabla^T\sqrt{N+\lambda}\boldsymbol{\xi}_i\boldsymbol{\xi}_i^T\sqrt{N+\kappa}\nabla)\mathbf{f}(\mathbf{x})\Big|_{\mathbf{x}=\bar{\mathbf{x}}}\right] \\ &= \frac{N+\lambda}{4(N+\lambda)}\left[(\nabla^T\sum_{i=1}^{2N}(\boldsymbol{\xi}_i\boldsymbol{\xi}_i^T)\nabla)\right]\mathbf{f}(\mathbf{x})\Big|_{\mathbf{x}=\bar{\mathbf{x}}} \\ &= \frac{N+\lambda}{4(N+\lambda)}\left[(\nabla^T 2\mathbf{P}_{\mathbf{x}}\nabla)\right]\mathbf{f}(\mathbf{x})\Big|_{\mathbf{x}=\bar{\mathbf{x}}} \\ &= \frac{1}{2}\left[(\nabla^T\mathbf{P}_{\mathbf{x}}\nabla)\right]\mathbf{f}(\mathbf{x})\Big|_{\mathbf{x}=\bar{\mathbf{x}}}\end{aligned} \tag{3.61}$$

Going from line 4 to 5 in (3.61), we used the fact that $\sum_{i=1}^{2N}\boldsymbol{\xi}_i\boldsymbol{\xi}_i^T = 2\mathbf{P}_{\mathbf{x}}$, because the range of summation is from 1 to $2N$. Inserting the result in (3.60), we obtain:

$$\bar{\mathbf{x}} = \mathbf{f}(\bar{\mathbf{x}}) + \frac{1}{2}\left[(\nabla^T\mathbf{P}_{\mathbf{x}}\nabla)\right]\mathbf{f}(\mathbf{x})\Big|_{\mathbf{x}=\bar{\mathbf{x}}} + \frac{1}{2(N+\lambda)}\sum_{i=1}^{2N}\left[+\frac{1}{4!}D^4_{\tilde{\boldsymbol{\xi}}_i}\mathbf{f} + \frac{1}{6!}D^6_{\tilde{\boldsymbol{\xi}}_i}\mathbf{f} + \dots\right] \tag{3.62}$$

Comparing (3.62) to (2.31) it can be seen that the approximation of posterior mean is accurate to the second order term in the Taylor series expansion and to the third order term for symmetrical distributions with zero odd moments. The magnitude of the error depends on the selection of the scaling parameter κ and higher order derivatives of the function $\mathbf{f}(\cdot)$.

The expression for approximated posterior covariance can be derived in a similar manner, starting from (2.34) in Chapter 2.

$$\begin{aligned}\mathbf{P}_{\mathbf{y}} = \mathbf{F}_{\bar{\mathbf{x}}}\boldsymbol{P}_{\mathbf{x}}\mathbf{F}_{\bar{\mathbf{x}}}^{T} &- \frac{1}{4}E\left[D^{2}_{\boldsymbol{\sigma}_{\boldsymbol{x}}}\mathbf{f}\right]E\left[D^{2}_{\boldsymbol{\sigma}_{\boldsymbol{x}}}\mathbf{f}\right]^{T} \\ &+ E\left[\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\frac{1}{i!j!}D^{i}_{\boldsymbol{\sigma}_{\boldsymbol{x}}}\mathbf{f}\left(D^{j}_{\boldsymbol{\sigma}_{\boldsymbol{x}}}\mathbf{f}\right)^{T}\right] \\ &- \left[\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\frac{1}{2i!2j!}E\left[D^{2i}_{\boldsymbol{\sigma}_{\boldsymbol{x}}}\mathbf{f}\right]E\left[D^{2j}_{\boldsymbol{\sigma}_{\boldsymbol{x}}}\mathbf{f}\right]^{T}\right] \end{aligned} \tag{3.63}$$

where $\mathbf{F}_{\bar{\mathbf{x}}} = \nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x})\big|_{\bar{\mathbf{x}}}$ is the Jacobian matrix of $\mathbf{f}(\mathbf{x})$ evaluated at $\mathbf{x} = \bar{\mathbf{x}}$. The second term in the above expression can be further simplified to read as follows

$$\frac{1}{4}E\left[D^{2}_{\boldsymbol{\sigma}_{\boldsymbol{x}}}\mathbf{f}\right]E\left[D^{2}_{\boldsymbol{\sigma}_{\boldsymbol{x}}}\mathbf{f}\right]^{T} = \frac{1}{4}\left[\mathbf{F}_{\bar{\mathbf{x}}}\boldsymbol{P}_{\mathbf{x}}\mathbf{F}_{\bar{\mathbf{x}}}^{T}\right]\left[\mathbf{F}_{\bar{\mathbf{x}}}\boldsymbol{P}_{\mathbf{x}}\mathbf{F}_{\bar{\mathbf{x}}}^{T}\right]^{T} \tag{3.64}$$

Using the results from derivation of posterior mean accuracy (3.62), the estimate for posterior covariance can be expressed as

$$\begin{aligned}\boldsymbol{P}_{\mathbf{y}} = \mathbf{F}_{\bar{\mathbf{x}}}\boldsymbol{P}_{\mathbf{x}}\mathbf{F}_{\bar{\mathbf{x}}}^{T} &- \frac{1}{4}\left[\mathbf{F}_{\bar{\mathbf{x}}}\boldsymbol{P}_{\mathbf{x}}\mathbf{F}_{\bar{\mathbf{x}}}^{T}\right]\left[\mathbf{F}_{\bar{\mathbf{x}}}\boldsymbol{P}_{\mathbf{x}}\mathbf{F}_{\bar{\mathbf{x}}}^{T}\right]^{T} \\ &+ \frac{1}{2N+\lambda}\sum_{k=1}^{2N}\left[\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\frac{1}{i!j!}D^{i}_{\boldsymbol{\sigma}_{\boldsymbol{x}k}}\mathbf{f}\left(D^{j}_{\boldsymbol{\sigma}_{\boldsymbol{x}k}}\mathbf{f}\right)^{T}\right] \\ &- \left[\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\frac{1}{4(2i)!(2j)!(N+\kappa)^{2}}\sum_{k=1}^{2N}\sum_{m=1}^{2N}D^{2i}_{\boldsymbol{\sigma}_{\boldsymbol{x}k}}f\left(D^{2j}_{\boldsymbol{\sigma}_{\boldsymbol{x}m}}f\right)^{T}\right] \end{aligned} \tag{3.65}$$

The full derivation of the UT posterior covariance can be found in the work of Julier (1997) and will not be given here. Similar applies to the derivations of higher order terms like skewness and kurtosis. Comparing the Equation (3.65) to (3.63) and (3.3.1) it is clear that the UT calculates the posterior covariance $\boldsymbol{P}_{\mathbf{y}}$ accurately to the first two terms of the Taylor series expansion. Furthermore, it has been shown by Julier and Uhlmann (1996) that errors in the higher order terms are consistently smaller than the ones in linearized case (as used in the extended Kalman filter).

In this derivation, β is assumed to be zero. If knowledge of the prior distribution is available, this parameter can be set to a value that will minimize the error in higher order moments.

3.3.2 Unscented Kalman filter

The unscented Kalman filter (UKF) can be constructed by applying the unscented transformation to the Kalman filtering framework in a straightforward manner. The resulting algorithm is summarized below using matrix form of unscented transform (Särkkä, 2008).

Algorithm 3.3: Unscented Kalman filter

1. Initialize state estimate by $\hat{\mathbf{x}}_{0|0} = \bar{\mathbf{x}}_0$, $\mathbf{P}_{0|0} = \bar{\mathbf{P}}_0$ and set $t = 1$.

2. Form sigma points and their weights according to (3.46) and (3.47).

$$\boldsymbol{\mathcal{X}}_t = \begin{bmatrix} \bar{\mathbf{x}}_t & \bar{\mathbf{x}}_t + \gamma\sqrt{\mathbf{P}_{\mathbf{x}_t}} & \bar{\mathbf{x}}_t - \gamma\sqrt{\mathbf{P}_{\mathbf{x}_t}} \end{bmatrix} \tag{3.66}$$

   where $\gamma = \sqrt{N + \lambda}$ and λ is defined by (3.48).

3. Prediction step: compute the predicted state mean and covariance.

$$\boldsymbol{\mathcal{X}}_{t|t-1} = \mathbf{f}(\boldsymbol{\mathcal{X}}_{t-1}, \mathbf{u}_{t-1}, \boldsymbol{\theta}) \tag{3.67}$$

$$\widehat{\mathbf{x}}_{t|t-1} = \sum_{i=0}^{2N} w_i^{(m)} \boldsymbol{\mathcal{X}}_{i,t|t-1} \tag{3.68}$$

$$\mathbf{P}_{t|t-1} = \sum_{i=0}^{2N} w_i^{(c)} (\boldsymbol{\mathcal{X}}_{i,t|t-1} - \widehat{\mathbf{x}}_{t|t-1})(\boldsymbol{\mathcal{X}}_{i,t|t-1} - \widehat{\mathbf{x}}_{t|t-1})^T \tag{3.69}$$

4. Update step: compute the predicted mean and measurement covariance and state cross covariance.

$$\boldsymbol{\mathcal{Y}}_{t|t-1} = \mathbf{g}(\boldsymbol{\mathcal{X}}_{t|t-1}, \mathbf{u}_t, \boldsymbol{\theta}) \tag{3.70}$$

$$\widehat{\mathbf{y}}_t = \sum_{i=0}^{2N} w_i^{(m)} \boldsymbol{\mathcal{Y}}_{t|t-1} \tag{3.71}$$

$$\mathbf{P}_{\widehat{y}_t} = \sum_{i=0}^{2N} w_i^{(c)} (\boldsymbol{\mathcal{Y}}_{t|t-1} - \widehat{\mathbf{y}}_t)(\boldsymbol{\mathcal{Y}}_{t|t-1} - \widehat{\mathbf{y}}_t)^T \tag{3.72}$$

$$\mathbf{P}_{\widehat{x}_t,\widehat{y}_t} = \sum_{i=0}^{2N} w_i^{(c)} (\boldsymbol{\mathcal{X}}_{i,t|t-1} - \widehat{\mathbf{x}}_{t|t-1})(\boldsymbol{\mathcal{Y}}_{t|t-1} - \widehat{\mathbf{y}}_t)^T \tag{3.73}$$

$$\mathbf{K}_t = \mathbf{P}_{\widehat{x}_t,\widehat{y}_t} \mathbf{P}_{\widehat{y}_t}^{-1} \tag{3.74}$$

$$\widehat{\mathbf{x}}_{t|t} = \widehat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \widehat{\mathbf{y}}_t) \tag{3.75}$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{P}_{\widehat{y}_t} \mathbf{K}_t^T \tag{3.76}$$

5. Set $t = t + 1$ and return to step 2.

3.3.3 Unscented Kalman smoother

The unscented Kalman smoother is an approximation based solution to discrete time optimal smoothing problem, where again the unscented transformation is used to form approximations of mean and covariance of the posterior random variable. The full derivation of the algorithm can be found in literature (Haykin, 2001) and only the final algorithm is presented here.

Algorithm 3.4: Unscented Kalman smoother

1. Run the unscented Kalman filter algorithm and store the state estimates.
2. Initialize the smoother algorithm by $\hat{\mathbf{x}}_{T|T}$, $\mathbf{P}_{T|T}$ and set $t = T - 1$.
3. Form sigma points according to (3.46) and (3.47).

$$\boldsymbol{\mathcal{X}}_t = \begin{bmatrix} \widehat{\mathbf{x}}_{t|t} & \widehat{\mathbf{x}}_{t|t} + \gamma\sqrt{\mathbf{P}_{t|t}} & \widehat{\mathbf{x}}_{t|t} - \gamma\sqrt{\mathbf{P}_{t|t}} \end{bmatrix} \tag{3.77}$$

4. Prediction step: compute the predicted state mean, predicted state covariance and the cross covariance.

$$\boldsymbol{\mathcal{X}}_{t+1|t} = \mathbf{f}(\boldsymbol{\mathcal{X}}_t, \mathbf{u}_t, \boldsymbol{\theta}) \tag{3.78}$$

$$\widehat{\mathbf{x}}_{t+1|t} = \sum_{i=0}^{2N} w_i^{(m)} \boldsymbol{\mathcal{X}}_{i,t+1|t} \tag{3.79}$$

$$\mathbf{P}_{t+1|t} = \sum_{i=0}^{2N} w_i^{(c)} (\boldsymbol{\mathcal{X}}_{i,t+1|t} - \widehat{\mathbf{x}}_{t+1})(\boldsymbol{\mathcal{X}}_{i,t+1|t} - \widehat{\mathbf{x}}_{t+1})^T \tag{3.80}$$

$$\mathbf{C}_{t+1} = \sum_{i=0}^{2N} w_i^{(c)} (\boldsymbol{\mathcal{X}}_{i,t|t-1} - \widehat{\mathbf{x}}_t)(\boldsymbol{\mathcal{X}}_{i,t+1|t} - \widehat{\mathbf{x}}_{t+1})^T \tag{3.81}$$

5. Update step: compute the smoother gain and smoothed mean and covariance.

$$\mathbf{D}_t = \mathbf{C}_{t+1} \left[\mathbf{P}_{t+1|t}\right]^{-1} \tag{3.82}$$

$$\widehat{\mathbf{x}}_{t|T} = \widehat{\mathbf{x}}_{t|t} + \mathbf{D}_t(\mathbf{x}_{t+1|T} - \widehat{\mathbf{x}}_{t+1|t}) \tag{3.83}$$

$$\mathbf{P}_{t|T} = \mathbf{P}_{t|t} + \mathbf{D}_t \left[\mathbf{P}_{t+1|T} - \mathbf{P}_{t+1|t}\right] \mathbf{D}_t^T \tag{3.84}$$

6. Set $t = t - 1$ and return to step 3.

The smoothed state distribution can then be approximated as:

$$p(\mathbf{x}_t|\mathbf{y}_{1:T}) \sim \mathcal{N}(\mathbf{x}_t^s, \mathbf{P}_{x_t}^s) \tag{3.85}$$

Additionaly, the joint distribution of $\mathbf{x}_{t+1}$ and $\mathbf{x}_t$ can be expressed as:

$$p(\mathbf{x}_{t+1}, \mathbf{x}_t | \mathbf{y}_{1:T}) \sim \mathcal{N}(\mathbf{x}_{t,t+1|T}, \mathbf{P}_{t,t+1|T}) \tag{3.86}$$

$$\mathbf{x}_{t,t+1|T} = \begin{pmatrix} \mathbf{x}_{t|T} \\ \mathbf{x}_{t+1|T} \end{pmatrix} \tag{3.87}$$

$$\mathbf{P}_{t,t+1|T} = \begin{pmatrix} \mathbf{P}_{t|T} & \mathbf{D}_t^T \mathbf{P}_{t+1|T} \\ \mathbf{P}_{t+1|t} \mathbf{D}_t^T & \mathbf{P}_{t+1|T} \end{pmatrix} \tag{3.88}$$

After backward run of the smoother algorithm we obtain the probability density functions $p(\mathbf{x}_t | \mathbf{Y}_T, \boldsymbol{\theta}') \sim \mathcal{N}(\mathbf{x}_{t|T}, \mathbf{P}_{t|T})$ and $p(\mathbf{x}_{t+1}, \mathbf{x}_t | \mathbf{Y}_T, \boldsymbol{\theta}') \sim \mathcal{N}(\mathbf{x}_{t,t+1|T}, \mathbf{P}_{t,t+1|T})$ for time index $t \in \{1, 2, \dots T\}$.

We have established a measure of approximation errors for posterior mean and covariance (c.f. (3.62) and (3.65)). However, when this approximations are used in the filtering and smoothing recursions, the approximated posterior distribution from one step is used as an exact prior in the subsequent step. Thus, generally, these errors accumulate, which deteriorates the estimation quality and can lead to divergence. This is a common problem of all approximation based approaches, but it may be more evident with the UT algorithms (even more in the EKF) due to relatively rough approximations used.

Example 3.2. State estimation using the unscented Kalman filter and smoother
The performance of the presented algorithms will be illustrated using a numerical example. The model that will be used here is adopted from literature and is often used as a benchmark model for state and parameter estimation (Arulampalam et al.*, 2002; Gordon* et al.*, 1993; Kitagawa, 1996).*

$$x_{t+1} = \frac{x_t}{2} + \frac{25 x_t}{1 + x_t^2} + 8 \cos(1.2t) + w_t \tag{3.89a}$$

$$y_t = \frac{x_t^2}{20} + v_t \tag{3.89b}$$

where w_t and v_t are mutually independent white Gaussian noise sequences, $w_t \sim \mathcal{N}(0, 10)$ and $w_t \sim \mathcal{N}(0, 1)$. The performance of the unscented Kalman filter and smoother is shown in Figure 3.5

3.4 Sequential Monte Carlo filtering

A particle filter, sometimes also called a sequential Monte Carlo filter, is a simulation based estimation technique. Sequential Monte Carlo filtering can be applied in numerous cases of nonlinear and non-Gaussian problems with multi-modal or heavy-tailed distribution, where the unscented Kalman filter (and also the extended Kalman filter) fails. Some

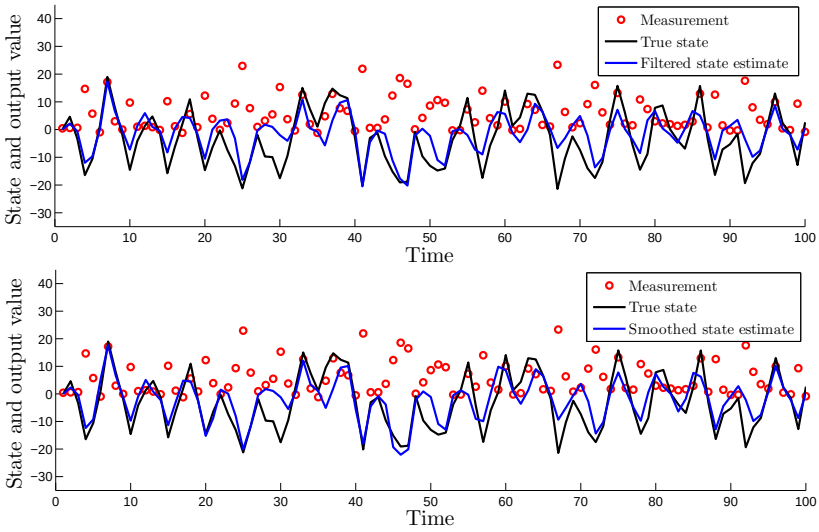


Figure 3.5: Unscented Kalman filter and unscented Kalman smoother results for example 3.2.

of these issues can be addressed by Gaussian sum filters (GSF) (Alspach and Sorenson, 1972), where posterior distribution is approximated by a sum of Gaussian distributions. However, due to its internal structure, which is the same as in the EKF, it suffers from the same shortcomings. The possible solution to non-Gaussian and nonlinear filtering problem can be provided by the Monte Carlo methods.

The key underlying idea behind Monte Carlo approach is to approximate the probability density function by a set of randomly selected samples (or particles, hence the name), and their corresponding weights. This may sound similar to the unscented transformation approach, but the difference between the ideas is that here the number of samples is not deterministic and relies on strong law of large numbers to correctly capture the distribution.

The posterior probability density function $p(\mathbf{x}_t|\mathbf{Y}_T)$ is then approximated by

$$p(\mathbf{x}_t|\mathbf{Y}_T) \approx \frac{1}{M}\sum_{i=1}^{M}\delta(\mathbf{x}_t - \mathbf{x}_{t|T}^{(i)}) \equiv \widehat{p}(\mathbf{x}_t|\mathbf{Y}_T) \tag{3.90}$$

where $\delta(\cdot)$ is a Dirac delta function and $\mathbf{x}_{t|T}^{(i)}$ are assumed to be i.i.d. drawn from $p(\mathbf{x}_t|\mathbf{Y}_T)$. When M is sufficiently large, $\widehat{p}(\mathbf{x}_t|\mathbf{Y}_T)$ approximates the true posterior probability density function $p(\mathbf{x}_t|\mathbf{Y}_T)$.

3.4.1 Particle approximations using perfect sampling

Let us assume that we have access to M i.i.d samples $\{\mathbf{x}^{(i)}\}_{i=1}^{M}$ from the probability density function $p(\mathbf{x})$. These samples will be used for calculating the estimates of the form (3.93). Even though the assumption we made is unrealistic, it will allow us to illustrate the basic principles of Monte Carlo based approximations.

Using the samples $\{\mathbf{x}^{(i)}\}_{i=1}^{M}$, the estimate of their probability density function can be formed as

$$\widehat{p}(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M}\delta(\mathbf{x}-\mathbf{x}^{(i)}) \tag{3.91}$$

Consequently, provided that the expectation of $f(\mathbf{x})$ exists,

$$E\left[f(\mathbf{x})\right] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \tag{3.92}$$

can be approximated according to

$$E\left[f(\mathbf{x})\right] \approx \widehat{E}\left[f(\mathbf{x})\right] = \int f(\mathbf{x})\widehat{p}(\mathbf{x})d\mathbf{x} = \frac{1}{M}\sum_{i=1}^{M} f(\mathbf{x}^{(i)}) \tag{3.93}$$

As M goes to infinity, according to the strong law of large numbers the estimates converge to the true expectation almost surely, i.e.,

$$\widehat{E}\left[f(\mathbf{x})\right] \xrightarrow[N\to\infty]{\text{a.s.}} E\left[f(\mathbf{x})\right] \tag{3.94}$$

Moreover, if the posterior covariance of $f(\mathbf{x})$ is bounded, that is

$$\mathbf{P}_{\mathbf{x}} = \text{Cov}\left[f(\mathbf{x})\right] = E\left[f(\mathbf{x})f(\mathbf{x})^T\right] < \infty \tag{3.95}$$

the central limit theorem can be applied

$$\sqrt{N}\left(\widehat{E}\left[f(\mathbf{x})\right] - E\left[f(\mathbf{x})\}\right]\right) \xrightarrow[M\to\infty]{\text{d}} \mathcal{N}(\mathbf{0}, \mathbf{P}_{\mathbf{x}}) \tag{3.96}$$

where $\xrightarrow[M\to\infty]{\text{d}}$ denotes convergence in distribution. This means that using a large amount of samples $\{\mathbf{x}^{(i)}\}_{i=1}^{M}$ we can estimate any expression of the form (3.92) using (3.93).

In practice, the assumptions underlying the above discussion are almost never valid, as it may prove to be impossible to obtain i.i.d. samples from $p(\mathbf{x})$. In order to use the above approach, we need to generate samples from arbitrary complex distributions.

3.4.2 Sampling algorithm

The fundamental problem in applying the particle approximations to state estimation is how to generate random samples $\{\mathbf{x}^{(i)}\}$ from too complex probability density function $p(\mathbf{x})$. There exist different algorithms for this and only one possible approach will be reviewed here. More information about the alternative algorithms along with a more detailed theoretical background can be found in Gilks *et al.* (1996); Liu (2001); Robert and Casella (2005).

The algorithms for generating random samples are based on the following idea. Since we cannot draw samples directly from $p(\mathbf{x})$, we draw samples from an alternative distribution $\pi(\mathbf{x})$. This distribution is referred to as the *sampling density* and the only restriction on $\pi(\mathbf{x})$ is that its support has to include the support of $p(\mathbf{x})$.

$$\forall \mathbf{x} \in \mathbf{R}^N, p(\mathbf{x}) > 0 \Rightarrow \pi(\mathbf{x}) > 0 \tag{3.97}$$

For each sample drawn from the sampling probability density function, $\mathbf{x}^{(i)} \sim \pi(\mathbf{x})$, the probability that it was in fact generated from the probability density function $p(\mathbf{x})$ can be calculated. This probability, referred to as the *acceptance probability*, can then be used to decide weather the sample $\mathbf{x}^{(i)}$ is in fact a member of $p(\mathbf{x})$ or not. The acceptance probability is usually expressed as a function $\gamma(\tilde{\mathbf{x}})$ and the whole process is defined as

$$p(\mathbf{x}) \propto \gamma(\mathbf{x})\pi(\mathbf{x}) \tag{3.98}$$

Depending on the way the acceptance probability is computed, different sampling strategies are obtained. In this dissertation we will limit ourselves to the *importance sampling*, more specifically to its extension called the *sampling importance resampling* (SIR) (Gordon *et al.*, 1993), which is in some communities better known as the *bootstrap* procedure (Smith and Gelfand, 1992).

Sampling importance resampling

The basic idea behind sampling importance resampling (SIR) is the *sequential importance sampling* (SIS) algorithm (Marshall, 1959). The SIR was introduced to alleviate some of the limitations of the SIS algorithm. To illustrate the basic concept, we will begin by introducing the importance sampling algorithm and later build on it to introduce the SIR algorithm. When discussing the importance sampling algorithms, the sampling density $\pi(\mathbf{x})$ is usually referred to as the *importance function*.

The expected value of the form (3.92) can be rewritten as

$$E\left[f(\mathbf{x})\right] = \int f(\mathbf{x}) \frac{p(\mathbf{x})}{\pi(\mathbf{x})} \pi(\mathbf{x}) d\mathbf{x} \tag{3.99}$$

where an estimate of $E\left[f(\mathbf{x})\right]$ can be obtained according to (3.93) by generating M samples $\{\mathbf{x}^{(i)}\}_{i=1}^{M}$ from $\pi(\mathbf{x})$ and forming

$$\widehat{E}\left[f(\mathbf{x})\right] = \frac{1}{M}\sum_{i=1}^{M}\gamma(\mathbf{x}^{(i)})f(\mathbf{x}^{(i)}) \tag{3.100}$$

where

$$\gamma(\mathbf{x}^{(i)}) = \frac{p(\mathbf{x}^{(i)})}{\pi(\mathbf{x}^{(i)})}, \quad i = 1, \ldots, N \tag{3.101}$$

which are referred to as the importance weights (or the importance ratios). In a special case when the normalization of $p(\mathbf{x})$ is known, the importance weights have a unit sum over samples. However, for a more common case when the normalizing factor of $p(\mathbf{x})$ is not known, the importance weights can only be evaluated up to a normalizing constant.

$$\gamma(\mathbf{x}^{(i)}) \propto \frac{p(\mathbf{x}^{(i)})}{\pi(\mathbf{x}^{(i)})}, \quad i = 1, \ldots, N \tag{3.102}$$

Therefore, in order to assure that $\sum_{i=1}^{M}\gamma(\mathbf{x}^{(i)}) = 1$, the importance weights have to be normalized

$$\tilde{\gamma}(\mathbf{x}^{(i)}) = \frac{\gamma(\mathbf{x}^{(i)})}{\sum_{j=1}^{M}\gamma(\mathbf{x}^{(j)})} \tag{3.103}$$

where $\tilde{\gamma}(\mathbf{x}^{(i)})$ are normalized importance weights. Using the above definitions, the importance sampling estimate is given as

$$\widehat{E}\left[f(\mathbf{x})\right] = \frac{1}{M}\sum_{i=1}^{M}\tilde{\gamma}(\mathbf{x}^{(i)})f(\mathbf{x}^{(i)}) \tag{3.104}$$

The importance sampling estimate given by (3.104) is *biased* and *consistent*. This means that the bias vanishes at a rate of $\mathcal{O}(M)$. More information about this and other related theoretical results regarding this algorithm can be found in Chen (2001); Jordan (1998).

From the definition above, the approximation for target density reads as follows

$$\widehat{p}(\mathbf{x}) = \sum_{i=1}^{M}\tilde{\gamma}(\mathbf{x}^{(i)})\delta(\mathbf{x} - \mathbf{x}^{(i)}) \tag{3.105}$$

where the importance weights include the information about how probable is the sample that was generated from the target density.

When considering the application of the above algorithm for recursive state estimation, it has to be repeated at every time instance. In this case, the SIS algorithm has a significant limitation, i.e., the variance of importance weights tends to increase with each

iteration (Doucet *et al.*, 2000) which has a harmful effect on the accuracy of the estimation. In order to obtain reasonable estimates, the variance should be close to zero. In practice, this effect can become evident after a few iterations of the algorithm by observing the importance weights. After some iterations, only a few particles will have non-zero importance weights. A large number of weights are thus effectively removed from the particle set because their weights make them insignificant. This phenomenon is referred to as *weight degeneracy* or *weight impoverishment* (Green, 1995; Liu and Chen, 1998).

One way to reduce (but not fully eliminate) this deficiency is to introduce an additional selection stage (*resampling*). In this stage, the particles with low importance weights are eliminated and replaced by the copies of particles with high weights. More formally, the selection scheme associates a number of children M_i to each particle $\{\mathbf{x}^{(i)}\}$, so that $\sum_{i=1}^{M} M_i = M$ (M_i may be zero). Effectively, we generate new particles $\{\tilde{\mathbf{x}}^{(i)}\}$, where we are limited to the samples $\{\mathbf{x}^{(i)}\}$. This is realized by resampling among the samples according to

$$\mathrm{P}(\tilde{\mathbf{x}}^{(i)} = \mathbf{x}^{(j)}) = \tilde{\gamma}(\mathbf{x}^{(j)}), \quad i = 1, \ldots, M \tag{3.106}$$

This idea was first introduced by Rubin (1988) and it presents a key step in estimating density functions recursively (Gordon *et al.*, 1993). The entire procedure can be summarized in a SIR algorithm as follows.

Sampling importance resampling

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

$$\gamma(\mathbf{x}^{(i)}) = \frac{p(\mathbf{x}^{(i)})}{\pi(\mathbf{x}^{(i)})}, \quad i = 1, \ldots, M \tag{3.107}$$

   and then normalize the importance weights.

$$\tilde{\gamma}(\mathbf{x}^{(i)}) = \frac{\gamma(\mathbf{x}^{(i)})}{\sum_{j=1}^{M} \gamma(\mathbf{x}^{(j)})}, \quad i = 1, \ldots, M \tag{3.108}$$

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

$$\Pr(\tilde{\mathbf{x}}^{(i)} = \mathbf{x}^{(j)}) = \tilde{\gamma}(\mathbf{x}^{(j)}), \quad i = 1, \ldots, M \tag{3.109}$$

After the selection and resampling steps we obtain M particles, distributed approximately according to the posterior distribution. Since the selection step favors creation of

multiple copies of the highest weight particles, many may end up having no children, and others with very high number of children. This is the inherent problem in resampling and as it introduces a dependence among particles. This may suggest that the resampling step by itself brings nothing and one could simply work with a fixed set of particles. However, it turns out, that the resampling step is necessary to be able to track movements in the target probability density function.

In many cases this dependence is critical and has to be removed. To achieve this, additional procedures have been proposed. Some of them include performing a single Markov chain Monte Carlo step (Andrieu *et al.*, 2003), *roughening* (Gordon *et al.*, 1993), or *jittering* (Fearnhead, 1998), or to resample from continuous approximations of the discrete probability density function, which is referred to as the *regularized particle filter* (Musso *et al.*, 2001).

Resampling step

The resampling step includes generating a new set of particles $\{\tilde{\mathbf{x}}^{(i)}\}_{i=1}^{M}$ with replacement from the old set of particles $\{\mathbf{x}^{(i)}\}_{i=1}^{M}$ in such a way that the probability of drawing $\mathbf{x}^{(i)}$ is given by its weight $\tilde{\gamma}(\mathbf{x}^{(i)})$.

$$\Pr(\tilde{\mathbf{x}}^{(i)} = \mathbf{x}^{(j)}) = \tilde{\gamma}(\mathbf{x}^{(j)}), \quad i = 1, \ldots, M \tag{3.110}$$

Several strategies on how to perform the resampling step have been proposed (Efron, 1982; Rubin, 1988; Smith and Gelfand, 1992), including *residual sampling* (Higuchi, 1997; Liu and Chen, 1998), *systematic resampling* (also named *minimal variance sampling*) (Carpenter *et al.*, 1999; Doucet *et al.*, 2001; Kitagawa, 1996) and *multinomial resampling* (Green, 1995; Smith and Gelfand, 1992). The manner in which the resampling step is conducted, influences the resampling quality and most importantly the computational complexity. Therefore, even though some authors report that the resampling algorithm does not significantly affect performance, the computational complexity greatly motivates the search for a good resampling procedure. In order to avoid going out of the scope of this dissertation, let us only illustrate the idea with a simple random resampling algorithm (Doucet *et al.*, 2000).

The idea behind *simple random resampling* is to select the new particles by comparing a random number, drawn form the uniform distribution $\mathcal{U}(0, 1)$ to the cumulative sum of the normalized importance weights. The procedure is presented in Figure 3.6.

3.4.3 Particle filter

The above discussion presents a necessary theoretical background for the construction of the particle filtering framework. In the sequel, the particle filter for the state estimation

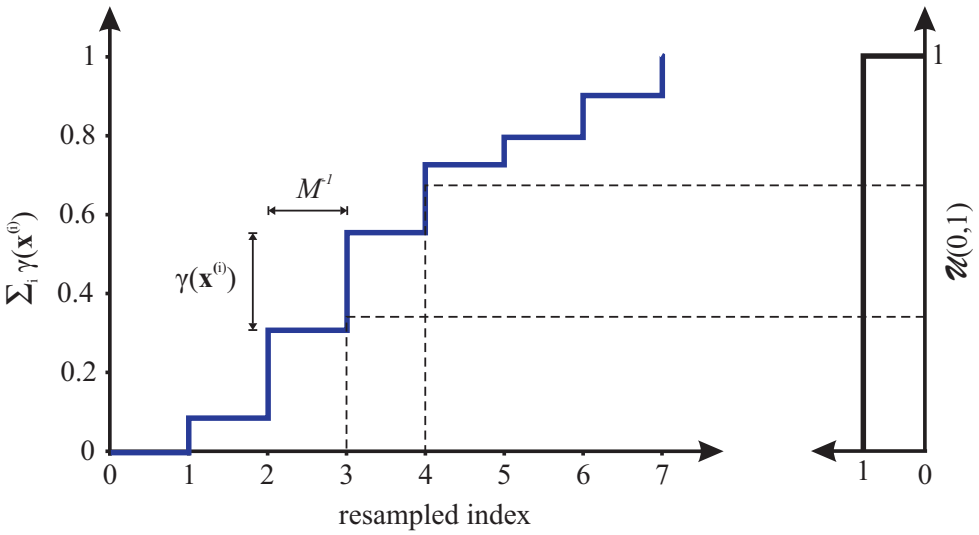


Figure 3.6: A set of new particles is obtained by first generating M uniformly distributed random numbers. Each number is then associated with a particle on the basis of cumulative sum of the normalized importance weights. In the figure above, two random numbers are generated and they correspond to selecting particle 3 and 4.

will be reviewed. This algorithm was first formally introduced by Gordon *et al.* (1993), while early ideas related to particle filter have emerged relatively early (Hammersley and Morton, 1954; Metropolis and Ulam, 1949). Later, important contributions have been made by Kitagawa (1996) and Isard and Blake (1998).

The particle filter will be derived using sampling importance resampling framework (Arulampalam *et al.*, 2002; Doucet *et al.*, 2000; Schön, 2006), where similar derivation could be done using any other resampling strategy. Consider the filtering problem and the unknown probability density function given by $p(\mathbf{x}_t|\mathbf{Y}_t)$.

Following the ideas from previous subsections, the appropriate sampling density $\pi(\mathbf{x}_t)$ has to be selected along with the appropriate acceptance probability. Using the Bayes' theorem and the Markov property, we can write

$$p(\mathbf{x}_t|\mathbf{Y}_t) = p(\mathbf{x}_t|\mathbf{y}_t, \mathbf{Y}_{t-1}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_{t-1})}{p(\mathbf{y}_t|\mathbf{Y}_{t-1})} \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_{t-1}) \tag{3.111}$$

Comparison to (3.98) leads to the following selection

$$\underbrace{p(\mathbf{x}_t|\mathbf{Y}_t)}_{p(\mathbf{x}_t)} \propto \underbrace{p(\mathbf{y}_t|\mathbf{x}_t)}_{\gamma(\mathbf{x}_t)} \underbrace{p(\mathbf{x}_t|\mathbf{Y}_{t-1})}_{\pi(\mathbf{x}_t)} \tag{3.112}$$

The above definition means that we can indeed generate samples from the target density using the procedure described in the previous section. Sampling importance resampling

algorithm dictates that the acceptance probabilities are calculated according to

$$\tilde{\gamma}(\mathbf{x}_{t|t-1}^{(i)}) = \frac{\gamma(\mathbf{x}_{t|t-1}^{(i)})}{\sum_{i=1}^{M}\gamma(\mathbf{x}_{t|t-1}^{(i)})} = \frac{p(\mathbf{y}_t|\mathbf{x}_{t|t-1}^{(i)})}{\sum_{i=1}^{M}p(\mathbf{y}_t|\mathbf{x}_{t|t-1}^{(i)})} \tag{3.113}$$

where $\mathbf{x}_{t|t-1}^{(i)} \sim p(\mathbf{x}_t|\mathbf{Y}_{t-1})$. Using the time update (Chapman-Kolmogorov) equation (3.15), these predicted particles can be calculated as

$$\begin{aligned}
\pi(\mathbf{x}_t) =& p(\mathbf{x}_t|\mathbf{Y}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{Y}_{t-1})d\mathbf{x}_{t-1} \\
&\approx \int p(\mathbf{x}_t|\mathbf{x}_{t-1})\sum_{i=1}^{M}\frac{1}{M}\delta\left(\mathbf{x}_{t-1}-\mathbf{x}_{t-1|t-1}^{(i)}\right)d\mathbf{x}_{t-1} \\
&= \sum_{i=1}^{M}\frac{1}{M}\int p(\mathbf{x}_t|\mathbf{x}_{t-1})\delta\left(\mathbf{x}_{t-1}-\mathbf{x}_{t-1|t-1}^{(i)}\right)d\mathbf{x}_{t-1} \\
&= \sum_{i=1}^{M}\frac{1}{M}p\left(\mathbf{x}_t|\mathbf{x}_{t-1|t-1}^{(i)}\right).
\end{aligned} \tag{3.114}$$

Equation (3.114) states, that the predicted particles $\{\mathbf{x}_{t|t-1}^{(i)}\}_{i=1}^{M}$ are simply generated by passing the particles from previous time instance $\{\mathbf{x}_{t-1|t-1}^{(i)}\}_{i=1}^{M}$ trough the underlying dynamical model. Furthermore, the acceptance probabilities $\tilde{w}(\mathbf{x}_{t|t-1}^{(i)})$ depend on the likelihood function $p(\mathbf{y}_t|\mathbf{x}_{t|t-1})$, which can be interpreted as a measure of how probable a particle is, given the current measurement. The particles with higher likelihood are more likely to be generated from the true density.

Using the sampling importance resampling algorithm, a new set of particles $\{\mathbf{x}_{t|t}^{(i)}\}_{i=1}^{M}$ that approximate $p(\mathbf{x}_t|\mathbf{Y}_t)$ is generated by

$$\Pr(\tilde{\mathbf{x}}_{t|t}^{(i)} = \mathbf{x}_{t|t-1}^{(j)}) = \tilde{\gamma}(\mathbf{x}_{t|t-1}^{(j)}), \quad i = 1, \ldots, M \tag{3.115}$$

Repeating this procedure yields the approximation

$$p(\mathbf{x}_t|\mathbf{Y}_t) \approx \sum_{i=1}^{M}\frac{1}{M}\delta\left(\mathbf{x}_t - \mathbf{x}_{t|t}^{(i)}\right) \tag{3.116}$$

which constitutes the particle filter. One iteration of the algorithm is illustrated in Figure 3.7.

Algorithm 3.5: Particle filter

1. Draw (sample) an initial set of M particles $\{\mathbf{x}_{1|0}^{(i)}\}$ from prior distribution $p(\mathbf{x}_1)$ and set $t = 1$.

2. Calculate the importance weights $w(\mathbf{x}_t^{(i)})$

$$\gamma(\mathbf{x}_{t|t-1}^{(i)}) = p\left(\mathbf{y}_t|\mathbf{x}_{t|t-1}^{(i)}\right), \quad i = 1, \ldots, M \tag{3.117}$$

and normalize them $\tilde{\gamma}(\mathbf{x}_{t|t-1}^{(i)}) = \gamma(\mathbf{x}_{t|t-1}^{(i)}) / \sum_{i=1}^{N} \gamma(\mathbf{x}_{t|t-1}^{(i)})$.

3. Draw M new particles with replacement.

$$\Pr\left(\tilde{\mathbf{x}_{t|t-1}}^{(i)} = \mathbf{x}_{t|t-1}^{(j)}\right) = \tilde{\gamma}(\mathbf{x}_{t|t-1}^{(j)}), \quad i = 1, \ldots, M \tag{3.118}$$

4. Predict new particles

$$\mathbf{x}_{t+1|t}^{(i)} = \mathbf{f}\left(\mathbf{x}_{t|t}^{(i)}\right), \quad i = 1, \ldots, M \tag{3.119}$$

5. Set $t = t + 1$ and go back to step 2.

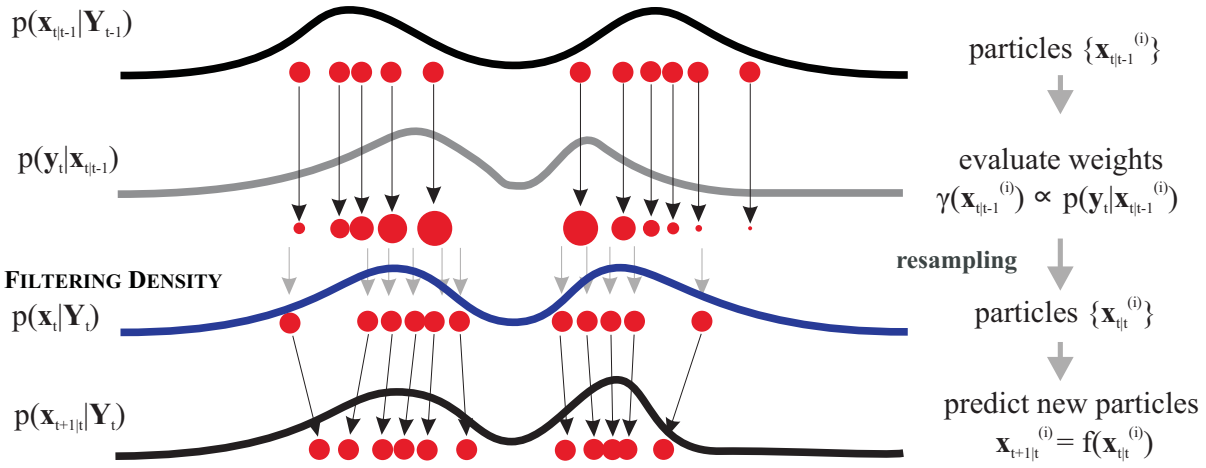


Figure 3.7: An iteration of a generic particle filter with resampling.

3.4.4 Particle smoother

In this section we address the problem of estimating the density function $p(\mathbf{x}_t|\mathbf{Y}_T)$, where $T > t$, using the sequential Monte Carlo approach. The approach presented in this section can indeed be applied to solve this problem and the algorithm is referred to as the *particle smoother*. Compared to the well-developed particle filtering theory, this problem has not received much attention (Schön, 2006).

In smoothing problem, the target density is given by smoothing density as

$$p\left(\mathbf{x}_{t+1}, \mathbf{x}_t\right) = p\left(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{Y}_T\right) \tag{3.120}$$

Next, we have to find an appropriate sampling density and the corresponding acceptance probabilities. According to the relations of joint distribution (3.3), we can write

$$p\left(\mathbf{x}_{t+1}, \mathbf{x}_t | \mathbf{Y}_T\right) = p\left(\mathbf{x}_t | \mathbf{x}_{t+1}, \mathbf{Y}_T\right) p\left(\mathbf{x}_{t+1} | \mathbf{Y}_T\right), \tag{3.121}$$

We use the Markov property

$$\begin{aligned} p\left(\mathbf{x}_t | \mathbf{x}_{t+1}, \mathbf{Y}_T\right) &= p\left(\mathbf{x}_t | \mathbf{x}_{t+1}, \mathbf{Y}_t, \mathbf{y}_{t+1}, \ldots, \mathbf{y}_T\right) \\ &= \frac{p\left(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T | \mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{Y}_t\right) p\left(\mathbf{x}_t | \mathbf{x}_{t+1}, \mathbf{Y}_t, \right)}{p\left(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T | \mathbf{x}_{t+1}, \mathbf{Y}_t\right)} \\ &= p\left(\mathbf{x}_t | \mathbf{x}_{t+1}, \mathbf{Y}_t\right) \\ &= \frac{p\left(\mathbf{x}_{t+1} | \mathbf{x}_t\right) p\left(\mathbf{x}_t | \mathbf{Y}_t\right)}{p\left(\mathbf{x}_{t+1} | \mathbf{Y}_t\right)} \end{aligned} \tag{3.122}$$

Combining the above equations then yields

$$\underbrace{p\left(\mathbf{x}_{t+1}, \mathbf{x}_t | \mathbf{Y}_T\right)}_{p(\mathbf{x}_{t+1}, \mathbf{x}_t)} = \underbrace{\frac{p\left(\mathbf{x}_{t+1} | \mathbf{x}_t\right)}{p\left(\mathbf{x}_{t+1} | \mathbf{Y}_t\right)}}_{\gamma(\mathbf{x}_{t+1}, \mathbf{x}_t)} \underbrace{p\left(\mathbf{x}_t | \mathbf{Y}_t\right) p\left(\mathbf{x}_{t+1} | \mathbf{Y}_T\right)}_{\pi(\mathbf{x}_{t+1}, \mathbf{x}_t)} \tag{3.123}$$

Using the above definition we can generate samples at time step t using the sampling density $\pi(\mathbf{x}_{t+1}, \mathbf{x}_t)$, and acceptance probabilities are calculated as

$$\gamma(\mathbf{x}_{t+1}, \mathbf{x}_t) = \frac{p\left(\mathbf{x}_{t+1} | \mathbf{x}_t\right)}{p\left(\mathbf{x}_{t+1} | \mathbf{Y}_t\right)}, \tag{3.124}$$

where $p\left(\mathbf{x}_{t+1} | \mathbf{x}_t\right)$ is defined by the underlying process model and $p\left(\mathbf{x}_{t+1} | \mathbf{Y}_t\right)$ is obtained using the result from the particle filter algorithm.

$$\begin{aligned} p\left(\mathbf{x}_{t+1} | \mathbf{Y}_t\right) &= \int p\left(\mathbf{x}_{t+1} | \mathbf{x}_t\right) p\left(\mathbf{x}_t | \mathbf{Y}_t\right) d\mathbf{x}_t \\ &= \int p\left(\mathbf{x}_{t+1} | \mathbf{x}_t\right) \sum_{i=1}^{M} \frac{1}{M} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(i)}\right) d\mathbf{x}_t \approx \sum_{i=1}^{M} \frac{1}{M} p\left(\mathbf{x}_{t+1} | \mathbf{x}_t^{(i)}\right) \end{aligned} \tag{3.125}$$

Resampling step can be done using the acceptance probabilities and can form approximations of target density function $p\left(\mathbf{x}_{t+1}, \mathbf{x}_t | \mathbf{Y}_T\right)$. The above expressions thus constitute the particle smoother algorithm (Tanizaki, 2001).

Algorithm 3.6: Particle smoother

1. Run the particle filter algorithm and store the filtered particles $\{\mathbf{x}_t^{(i)}\}_{i=1}^{M}, t = 1, \ldots, T$.

2. Initialize the smoothed particles with importance weights at time T.

$$\mathbf{x}_{T|T}^{(i)} = \mathbf{x}_T^{(i)}, \quad \gamma(\mathbf{x}_{T|T}^{(i)}) = 1/M, \quad i = 1, \ldots, M \tag{3.126}$$

3. Calculate the importance weights $\gamma(\mathbf{x}_{t|T}^{(i)})$

$$\gamma(\mathbf{x}_{t|T}^{(i)}) = \frac{p\left(\mathbf{x}_{t+1|T}^{(i)}|\mathbf{x}_{t|T}^{(i)}\right)}{\sum_{j=1}^{M} p\left(\mathbf{x}_{t+1|T}^{(i)}|\mathbf{x}_{t|T}^{(j)}\right)}, \quad i = 1, \ldots, M \tag{3.127}$$

and normalize them $\tilde{\gamma}(\mathbf{x}_{t|T}^{(i)}) = \gamma(\mathbf{x}_{t|T}^{(i)}) / \sum_{i=j}^{M} \gamma(\mathbf{x}_{t|T}^{(j)})$.

4. Draw M new particles with replacement.

$$\Pr\left(\left(\mathbf{x}_{t+1|T}^{(i)}, \mathbf{x}_{t|T}^{(i)}\right) = \left(\mathbf{x}_{t+1|T}^{(j)}, \mathbf{x}_{t|T}^{(j)}\right)\right) = \tilde{\gamma}(\mathbf{x}_{t|T}^{(i)}), \quad i = 1, \ldots, M \tag{3.128}$$

5. Set $t = t - 1$ and go back to step 3.

Example 3.3. State estimation: particle filter and particle smoother

With the aim to compare the sequential Monte Carlo to the unscented approach, we again use the model from Example 3.2 (3.89). The basic particle filter and smoother algorithms were used and were initialized with $M = 100$ samples. The results are shown in Figure 3.8.

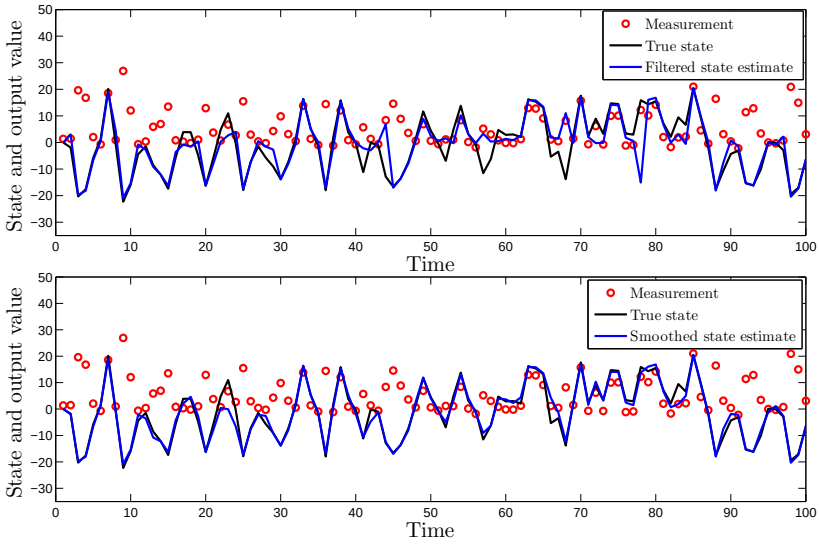
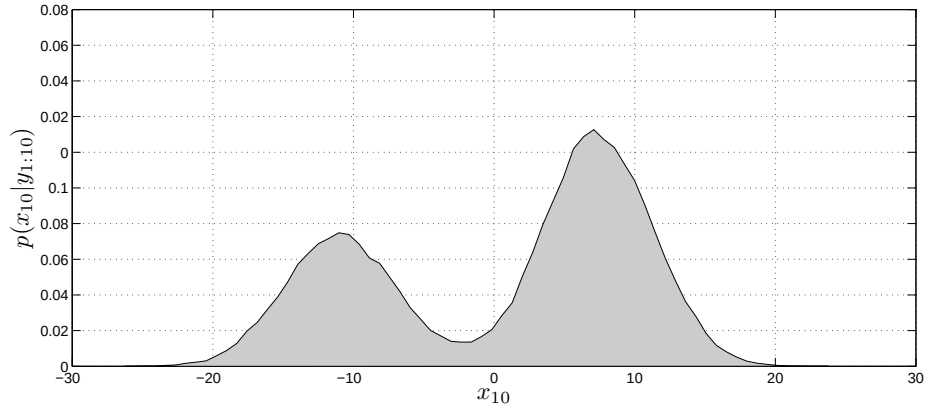


Figure 3.8: Particle filter and particle smoother results for example 3.2.

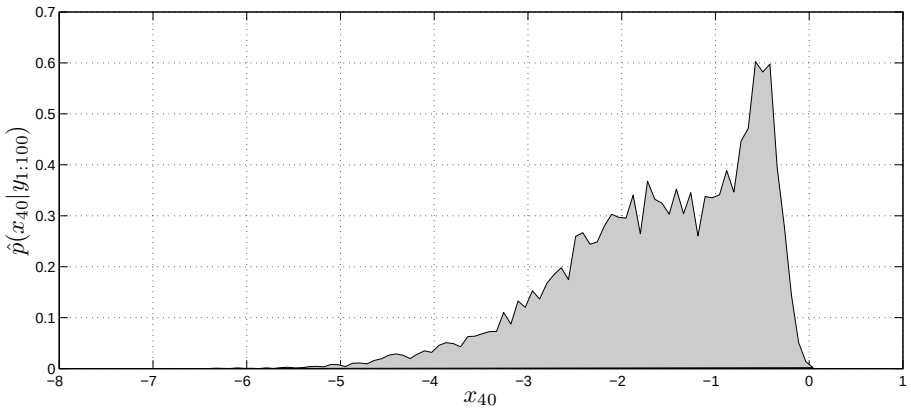
It can be seen that the state estimates are more accurate than in the ones obtained using the unscented Kalman filter and the unscented Kalman smoother. The comparison of the algorithms in terms of MSE is given in Table 3.1. The reason for the relatively poor performance of the unscented Kalman filter and smoother can be found in the nonlinear

Table 3.1: State estimation of nonlinear model (3.89) using the unscented Kalman filter and the sequential Monte Carlo algorithms.

Algorithm	**MSE**
unscented Kalman filter (Algortihm 3.3)	61.78
unscented Kalman smoother (Algortihm 3.4)	45.99
particle filter (Algortihm 3.5)	37.72
particle smoother (Algortihm 3.6)	12.57



(a) Filtering estimate of the probability density function from the particle filter algorithm



(b) Smoothing estimate of the probability density function from the particle smoother algorithm

Figure 3.9: Filtering and smoothing probability density functions.

nature of the model. The individual state probability density functions, shown in Figure (3.3), clearly indicate that the densities are not Gaussian and thus cannot be accurately approximated using the unscented transformation.

3.5 Chapter summary

This chapter reviews the problem of state filtering and smoothing in state-space models using Bayesian optimal filtering theory with the aim to prepare the floor for the parameter estimation problem in the next chapter.

First, basic theorems were presented followed by the optimal Bayesian filtering and smoothing theory. Next, we have reviewed three algorithms tailored for different model classes. In the case of linear and Gaussian models an explicit solution is derived, known as the Kalman filter and smoother.

On the other hand, nonlinear systems pose a far greater challenge, mainly arising from the problem of calculating the posterior distribution of random variable that undergoes a nonlinear transformation. This problem was approached from two perspectives: using the unscented transformation and sequential Monte Carlo methodology.

Firstly, the solution using the unscented transformation was presented. This is an approximate solution which uses a set of deterministically chosen points to approximate the probability distribution. These points are then transformed by the nonlinear function and the first two moments of the posterior distribution can be explicitly formed using the transformed points. The approximation error depends on the type of nonlinear functions and properties of random components. Unfortunately, the exact relation between them and the magnitude of the error is still unclear.

Secondly, the sequential Monte Carlo filtering theory provides a more robust solution to optimal state estimation of nonlinear and non-Gaussian systems. However, in comparison to the unscented Kalman filter, the computational load required to achieve this accuracy may be extremely large, especially in high dimensional problems.

These are only three possible algorithms that can be used for state estimation. More details on alternative algorithms for this purpose can be found in the works in the reference list. The particular algorithms discussed here have been selected because they represent an extremely useful set of tools for solving filtering problems: the Kalman filter for linear models, the unscented Kalman filter for high-dimensional problems where nonlinearities are not too "severe" and finally, the particle filter for highly nonlinear and non-Gaussian models.

4 Maximum likelihood parameter estimation

In this chapter we will address the problem of estimating unknown model parameters of the nonlinear state-space models (as defined in Chapter 2). Estimating the state-space models is challenging because the internal system states are not directly observed and therefore all the information about them has to be inferred form the measured data. We addressed the estimation of this state sequence in the previous chapter, but we made an assumption that we know the true values of model parameters. As this is usually not the case, there is a need for a systematic approach that allows both the estimation of system states and unknown model parameters.

In the scope of this dissertation, unknown model parameters will be treated as constants. This type of problem is usually solved by constructing an appropriate point estimator for the vector of unknown model parameters $\boldsymbol{\theta}$. A point estimator $\widehat{\boldsymbol{\theta}}$ is a single value statistic that is used to infer the value of an unknown parameter based on the observed data. Because the estimator is a function of the data, it is itself a random variable. The point estimate of a parameter vector $\boldsymbol{\theta}$ is thus a particular realization of this random variable, computed from the given data.

Different estimators can be constructed in this manner, depending on the definition of the criterion of optimality. An appropriate method in terms of the estimator properties is the *maximum likelihood*, which was developed by a R. A. Fisher (Fisher, 1912).

The chapter will first present an intuitive solution to the system identification problem that will sketch the ideas behind the proposed algorithm. The solution is based on a specific estimator called the *maximum likelihood estimator*, which will be presented and analyzed in terms of the estimation properties. With introduction of the maximum likelihood estimation we will fulfill all the prerequisites for introduction of the *expectation-maximization* algorithm, which is the core of our identification procedure. Three different algorithms based on the expectation-maximization approach will be presented, where each will be tailored to a specific family of models.

4.1 Motivation and intuitive approach

In order to provide a better understanding of the problem, let us adopt intuition and try to find a solution to the system identification problem. Consider a linear state-space model

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t \tag{4.1a}$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t + \mathbf{v}_t \tag{4.1b}$$

If the system states $\mathbf{x}_t$ were known, one could simply estimate the model parameters (matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$) by linear regression.

$$\begin{bmatrix} \widehat{\mathbf{A}} & \widehat{\mathbf{B}} \\ \widehat{\mathbf{C}} & \widehat{\mathbf{D}} \end{bmatrix} = \left(\sum_{i=1}^{T} \begin{bmatrix} \mathbf{x}_{t+1}\mathbf{x}_t^T & \mathbf{x}_{t+1}\mathbf{u}_t^T \\ \mathbf{y}_t\mathbf{x}_t^T & \mathbf{y}_t\mathbf{u}_t^T \end{bmatrix} \right) \left(\sum_{i=1}^{T} \begin{bmatrix} \mathbf{x}_t\mathbf{x}_t^T & \mathbf{x}_t\mathbf{u}_t^T \\ \mathbf{u}_t\mathbf{x}_t^T & \mathbf{u}_t\mathbf{u}_t^T \end{bmatrix} \right)^{-1} \tag{4.2}$$

Assume that the state sequence used in regression was just an estimate (e.g. using the Kalman filter) conditioned on some parameter values. Then the new estimated parameters could be used to obtain new estimates of the system states. It would make sense to repeat the regression step using a new state sequence, which would in turn produce new estimates of the model parameters. Our solution would then be to iterate these steps and gradually improve the parameter estimates and state estimates and eventually obtain the optimal estimates.

The same logic is applied in the expectation-maximization algorithm. The EM algorithm relies on the theoretical background which guarantees that the estimates will converge to some optimal values.

4.2 Maximum likelihood estimator

Suppose $\mathbf{x}$ is a random variable with probability density function $p_{\boldsymbol{\theta}}(\mathbf{x})$, where $\boldsymbol{\theta}$ is a vector of unknown parameters and $p_{\boldsymbol{\theta}}(\cdot)$ denotes the probability density function $p(\cdot)$ which is parameterized by parameter vector $\boldsymbol{\theta}$. Let $\mathbf{X}_T = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ be the set of observed values. The probability density function of $\mathbf{X}_T$ is

$$p_{\boldsymbol{\theta}}(\mathbf{X}_T) = p_{\boldsymbol{\theta}}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) \tag{4.3}$$

The probability density function $p_{\boldsymbol{\theta}}(\mathbf{X}_T)$ is a deterministic function of $\boldsymbol{\theta}$ as $\mathbf{X}_t$ is fixed at measured values and is referred to as the *likelihood function*. A reasonable estimator for $\boldsymbol{\theta}$ could then be to select the values in such a way that the observed realization $\mathbf{X}_T$ becomes as likely as possible.

Maximum likelihood estimator

Maximum likelihood estimator for unknown parameters is defined by a function

$$\widehat{\boldsymbol{\theta}}_{ML}(\mathbf{X}_T) = \arg\max_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\mathbf{X}_T) \tag{4.4}$$

where the maximization is performed with respect to $\boldsymbol{\theta}$ and for a fixed $\mathbf{X}_T$.

Rather than (4.4) it is often convenient to operate with the log-likelihood function.

$$L(\boldsymbol{\theta}) = \log p_{\boldsymbol{\theta}}(\mathbf{X}_T) \tag{4.5}$$

Since logarithmic function is monotonically increasing, maximizing the likelihood function is the same as maximizing its logarithm

$$\widehat{\boldsymbol{\theta}}_{ML}(\mathbf{X}_T) = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \tag{4.6}$$

Example 4.1. *Let $X_T = \{x_1, x_2, \ldots, x_T\}$ be realization of conditionally independent random samples from Gaussian distribution with mean μ and variance σ^2, where both μ and σ^2 are unknown. Since random variables are independent, the likelihood function for sample size of n is given by*

$$p_{\mu,\sigma^2}(X_T) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2} \tag{4.7}$$

Consequently,

$$\begin{aligned} L_{\mu,\sigma^2}(X_T) &= \log\left(p_{\mu,\sigma^2}(X_T)\right) \\ &= -\frac{n}{2}\log\left(2\pi\sigma^2\right) - -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2 \end{aligned} \tag{4.8}$$

The derivatives with respect to unknown parameters read

$$\frac{\partial L_{\mu,\sigma^2}(X_T)}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu) = 0 \tag{4.9a}$$

$$\frac{\partial L_{\mu,\sigma^2}(X_T)}{\partial\left(\sigma^2\right)} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}(x_i-\mu)^2 = 0 \tag{4.9b}$$

Solving the above equations provides the maximum likelihood estimators for mean and variance as

$$\widehat{\mu}_{ML} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.10a}$$

$$\widehat{\sigma}^2_{ML} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \widehat{\mu}_{ML})^2 \tag{4.10b}$$

Equations (4.10) represent a well-known point estimators for mean and variance.

4.2.1 Relation to maximum a posteriori (MAP) estimation

A maximum a posteriori estimator is another well-known point estimator that arises from the Bayesian approach. The solution is similar to the ML estimate, but conceptually different. The comparison of the two approaches continuously fuels many debates. In the Bayesian approach, the parameters are treated as random variables. From the observation of other random variables (e.g., measured outputs), which are correlated to the parameters, we can infer the information about the parameters.

Consider the known prior distribution of the unknown parameter $\boldsymbol{\theta}$ to be $p(\boldsymbol{\theta})$ and observed data likelihood function $p(\mathbf{X}_T|\boldsymbol{\theta})$. Using Bayes' rule, the conditional probability density function $\boldsymbol{\theta}$, given the observed data, is

$$p(\boldsymbol{\theta}|\mathbf{X}_T) = \frac{p(\mathbf{X}_T|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{X}_T)} \propto p(\mathbf{X}_T|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{4.11}$$

The posterior probability density function of $\boldsymbol{\theta}$ is thus proportional to the likelihood function, multiplied by the prior.

The MAP estimate is a point estimate of the posterior distribution mean value and can be written as

$$\widehat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}}\{p(\mathbf{X}_T|\boldsymbol{\theta})p(\boldsymbol{\theta})\} \tag{4.12}$$

In a case of uniform prior probability density function, the MAP estimate is the same as the ML estimate. For a general prior probability density function, the MAP estimate is asymptotically close to the ML estimate if the number of samples T is large and the prior distribution has an insignificant influence.

4.2.2 Properties of the maximum likelihood estimator

The method of maximum likelihood is often used as the parameter estimation method, mostly owing to the fact that it produces estimators which have good statistical properties.

Asymptotic unbiasedness

Bias of the estimator reflects the fact that the estimator $\widehat{\boldsymbol{\theta}}$ has no systematic tendency to produce estimates different from the true parameter value $\boldsymbol{\theta}^*$. Of course this is a very desirable property of an estimator and estimator that guarantees this is called *unbiased*. Formally, we can write this as

$$E\left[\widehat{\boldsymbol{\theta}}(\mathbf{X}_T)\right] = \boldsymbol{\theta}^* \tag{4.13}$$

where $\mathbf{X}_T$ has a probability density function $p_{\boldsymbol{\theta}^*}(\mathbf{X}_T)$. The difference $E\left[\widehat{\boldsymbol{\theta}}(\mathbf{X}_T)\right] - \boldsymbol{\theta}^*$ is called *bias* and if it is not equal to zero, the estimator is referred to as *biased*. A less

strict criterion for the estimator quality is the *asymptotic unbiasedness*, which states that the estimator produces unbiased estimate in the limit when the number of samples goes to infinity.

$$\lim_{T\to\infty}\left(E\left[\widehat{\boldsymbol{\theta}}(\mathbf{X}_T)\right]-\boldsymbol{\theta}^*\right)=0 \tag{4.14}$$

Example 4.2. *Consider the estimator for variance of the Gaussian distribution given in Example 4.1.*

$$\widehat{\sigma}^2_{ML}=\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\widehat{\mu}_{ML}\right)^2 \tag{4.15}$$

The expected value of this estimator is

$$E\left[\widehat{\sigma}^2_{ML}\right]=\frac{1}{n}\sum_{i=1}^{n}E\left(x_i-\widehat{\mu}_{ML}\right)^2=\frac{1}{n}\sum_{i=1}^{n}\mathrm{Var}\left(x_i-\widehat{\mu}_{ML}\right) \tag{4.16}$$

where

$$\begin{aligned}E\left(x_i-\widehat{\mu}_{ML}\right)^2&=E\left(\frac{n-1}{n}x_i-\frac{1}{n}\sum_{j\neq i}x_j\right)^2=\frac{(n-1)^2}{n^2}E(x_i)^2+\frac{1}{n^2}\sum_{j\neq i}E(x_j)^2\\&=\left[\frac{(n-1)^2}{n^2}+\frac{n-1}{n^2}\right]\sigma^2=\frac{n-1}{n}\sigma^2\end{aligned} \tag{4.17}$$

and thus

$$E\left[\widehat{\sigma}^2_{ML}\right]=\frac{1}{n}\sum_{i=1}^{n}E\left(x_i-\widehat{\mu}_{ML}\right)^2=\frac{1}{n}\sum_{i=1}^{n}\frac{n-1}{n}\sigma^2=\frac{n-1}{n}\sigma^2 \tag{4.18}$$

We can see that $\widehat{\sigma}^2_{ML}$ is a biased estimator for σ^2, but also, as n goes to infinity, the bias goes to zero, thus maximum likelihood estimator is asymptotically unbiased estimator for variance of Gaussian distribution.

It can be shown that this holds more generally, as under mild conditions on the distribution of the random variable the maximum likelihood estimator is asymptotically unbiased (Dekking *et al.*, 2005; van de Geer, 2000).

Strong consistency

The estimator is said to be consistent (or asymptotically consistent) if, as a number of samples T goes to infinity, the estimator $\widehat{\boldsymbol{\theta}}$ converges in probability to its true value $\boldsymbol{\theta}^*$. Maximum likelihood estimator is consistent under certain conditions (Newey and McFadden, 1994).

$$\widehat{\boldsymbol{\theta}}_{ML}(\mathbf{X}_T)\xrightarrow{\mathrm{p}}\boldsymbol{\theta}^* \tag{4.19}$$

where $\xrightarrow{\text{p}}$ denotes convergence in probability. Furthermore, it can be shown that under slightly stronger conditions, the maximum likelihood estimator also converges almost surely (Newey and McFadden, 1994), which is the required condition for strong consistency.

$$\widehat{\boldsymbol{\theta}}_{ML}(\mathbf{X}_T) \xrightarrow{\text{a.s.}} \boldsymbol{\theta}^* \tag{4.20}$$

Asymptotic efficiency

Asymptotic efficiency refers to the case when the asymptotic variance of the estimates equals the inverse Fisher information matrix, which is the minimum possible variance, defined by the Cramér-Rao lower bound.

Cramér-Rao inequality

Let $\widehat{\boldsymbol{\theta}}(\mathbf{X}_T)$ be an estimator of $\boldsymbol{\theta}$, such that $E\left[\widehat{\boldsymbol{\theta}}(\mathbf{x}_T)\right] = \boldsymbol{\theta}^*$ and assume $\mathbf{X}_T$ has a probability density function $p_{\boldsymbol{\theta}^*}(\mathbf{X}_T)$. Then

$$E\left[\widehat{\boldsymbol{\theta}}(\mathbf{X}_T) - \boldsymbol{\theta}^*\right]\left[\widehat{\boldsymbol{\theta}}(\mathbf{X}_T) - \boldsymbol{\theta}^*\right]^T \geq \mathcal{I}(\boldsymbol{\theta}^*)^{-1} \tag{4.21}$$

where

$$\begin{aligned}\mathcal{I}(\boldsymbol{\theta}^*) &= E\left[\frac{\partial}{\partial \boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{X}_T)\right]\left[\frac{\partial}{\partial \boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{X}_T)\right]^T\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} \\ &= -E\left[\frac{\partial^2}{\partial \boldsymbol{\theta}^2} \log p_{\boldsymbol{\theta}}(\mathbf{X}_T)\right]\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}\end{aligned} \tag{4.22}$$

The matrix $\mathcal{I}(\boldsymbol{\theta}^*)$ is known as the Fisher information matrix.

In mathematical terms this means that when the number of samples T goes to infinity, the random variable $\widehat{\boldsymbol{\theta}}_{ML}(\mathbf{X}_T)$ tends to $\boldsymbol{\theta}^*$ with probability 1 and random variable $\sqrt{T}\left[\widehat{\boldsymbol{\theta}}_{ML}(\mathbf{X}_T) - \boldsymbol{\theta}^*\right]$ converges in distribution to the normal distribution with zero mean and covariance matrix given by Cramér-Rao lower bound.

$$\sqrt{T}\left[\widehat{\boldsymbol{\theta}}_{ML}(\mathbf{X}_T) - \boldsymbol{\theta}^*\right] \xrightarrow{\text{d}} \mathcal{N}(0, \mathcal{I}(\boldsymbol{\theta}^*)) \tag{4.23}$$

where $\xrightarrow{\text{d}}$ denotes convergence in distribution. This classical result was given by Cramer (1946), while a detailed proof can also be found in book by Grenander and Miller (2007).

4.2.3 Likelihood function for dynamical models

Consider a dynamic state-space model, where $\mathbf{Y}_T$ are measured outputs, $\mathbf{X}_T$ is the unobserved sequence of system states and $\boldsymbol{\theta}$ is vector of unknown model parameters. A straightforward way to define the maximum likelihood parameter estimator is

$$\widehat{\boldsymbol{\theta}}_{ML}(\mathbf{Y}_T) = \arg\max_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\mathbf{Y}_T) \tag{4.24}$$

where the data likelihood function can be expressed using chain rule

$$p_{\boldsymbol{\theta}}(\mathbf{Y}_T) = p_{\boldsymbol{\theta}}(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T) = p_{\boldsymbol{\theta}}(\mathbf{y}_1) \prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{Y}_{t-1}) \tag{4.25}$$

However, it is convenient to consider the log-likelihood function

$$L(\boldsymbol{\theta}) = \log p_{\boldsymbol{\theta}}(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T) = \sum_{t=2}^{T} \log p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{Y}_{t-1}) + \log p_{\boldsymbol{\theta}}(\mathbf{y}_1) \tag{4.26}$$

The maximum likelihood estimator is

$$\widehat{\boldsymbol{\theta}}_{ML}(\mathbf{Y}_T) = \arg\max_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\mathbf{Y}_T) = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \tag{4.27}$$

A closer look at the expression $p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{Y}_{t-1})$ in (4.26) reveals that it depends on system states. Indeed

$$p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{Y}_{t-1}) = \int p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t) p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{Y}_{t-1}) d\mathbf{x}_t \tag{4.28}$$

The formulation of the above integral is problematic and in general case no closed form solutions exist. The problem of calculating the maximum of the data likelihood function of dynamical systems is the main topic of this chapter and we will demonstrate how it can be solved in Section 4.4.

Relation to the prediction error method

The maximum likelihood method can be related to another popular method for parameter estimation in dynamical models, named the *prediction error method* (Ljung, 1999). In the prediction error method, the estimate of the model parameters is obtained by solving the following optimization problem

$$\widehat{\boldsymbol{\theta}}_{PE} = \arg\min_{\boldsymbol{\theta}} V_T(\boldsymbol{\theta}) \tag{4.29}$$

where

$$V_T(\boldsymbol{\theta}) = \frac{1}{T} \sum_{t=1}^{T} \ell\left(\epsilon_t(\boldsymbol{\theta})\right) \tag{4.30}$$

Here $\epsilon_t(\boldsymbol{\theta}) = \mathbf{y}_t - \widehat{\mathbf{y}}_t(\boldsymbol{\theta})$ is the prediction error and $\ell(\epsilon_t(\boldsymbol{\theta}))$ is a suitably chosen positive function. If this function is defined as

$$\ell(\epsilon_t(\boldsymbol{\theta})) = -\log p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{Y}_{t-1}) \tag{4.31}$$

the prediction error method becomes equivalent to the maximum likelihood method. Indeed, in the scalar Gaussian case and a common choice of $\ell(x) = x^2$, it has already been established that the prediction error estimate is asymptotically equal to maximum likelihood estimate (Ninnes, 2009).

4.3 Expectation-maximization algorithm

The Expectation-Maximization (EM) algorithm is an efficient iterative procedure for computing the maximum likelihood estimates in the presence of missing data. The paper by Dempster *et al.* (1977) is usually considered as a seminal work, but similar methods can also be found earlier, such as the Baum-Welsch method for estimation in discrete valued hidden Markov models (Baum *et al.*, 1970).

The EM algorithm is widely used in the area of applied statistics. Numerous successful applications can be found in the literature, ranging from speech recognition (Digalakis *et al.*, 1993), image processing (Borran and Aazhang, 2002) and data mining (Witten and Frank, 2005). More recently, the EM algorithm has been successfully applied to the system identification of different system setups (Andrieu *et al.*, 2004; Gibson and Ninness, 2005; Shumway and Stoffer, 2005). Following the ideas laid down in this direction, we will demonstrate how the EM algorithm can be successfully applied to the maximum likelihood parameter estimation problem.

As soon as the system output depends also on the vector of the system states $\mathbf{X}_T$, the direct maximization of the log-likelihood function is not possible since the computation requires knowledge of the state vector (Equation 4.28). In order to evaluate the data log-likelihood function $L(\boldsymbol{\theta})$, it is clear that besides maximization of the function with respect to $\boldsymbol{\theta}$, the estimation of state vector ($\mathbf{X}_T$) is also required.

The expectation-maximization algorithm can solve the ML estimation problem in the case of incomplete or missing data. Therefore, if the states $\mathbf{X}_T$ are considered as missing data, this algorithm can be successfully deployed to solve the system identification problem. Consider an extension to (4.24).

$$\widehat{\boldsymbol{\theta}}_{ML}(\mathbf{X}_T, \mathbf{Y}_T) = \arg\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{X}_T, \mathbf{Y}_T) \tag{4.32}$$

The EM algorithm then solves the problem of simultaneously estimating system states and model parameters by alternating between two steps. First, it approximates the likelihood

function with its expected value over the missing data (E-step), and secondly maximizes the likelihood function w.r.t. $\boldsymbol{\theta}$ (M-step). A short overview of the algorithm will be presented, while a more detailed explanation can be found in Gibson and Ninness (2005); Haykin (2001).

It is worth stressing that the variable for the missing data should be chosen carefully. The choice should guarantee that if the data were known, solving (4.32) is indeed simple. In other case it can lead to a harder problem than the initial one.

4.3.1 The EM algorithm derivation

To include the state vector into the likelihood function we can use the chain rule (Dekking *et al.*, 2005) to derive the relation between the classical likelihood function and the joint likelihood function of the measurement data and the system states:

$$p_{\boldsymbol{\theta}}(\mathbf{Y}_T, \mathbf{X}_T) = p_{\boldsymbol{\theta}}(\mathbf{X}_T|\mathbf{Y}_T)p_{\boldsymbol{\theta}}(\mathbf{Y}_T) \tag{4.33}$$

Taking logarithms of the Equation 4.33 and rearranging it, we can write the following relation between the classical and the complete data log-likelihood function:

$$\log p_{\boldsymbol{\theta}}(\mathbf{Y}_T) = \log p_{\boldsymbol{\theta}}(\mathbf{X}_T, \mathbf{Y}_T) - \log p_{\boldsymbol{\theta}}(\mathbf{X}_T|\mathbf{Y}_T) \tag{4.34}$$

Taking the expectation of (4.34) with respect to $p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)$, where $\boldsymbol{\theta}'$ is any fixed value of the parameter vector, we obtain

$$\begin{aligned} E_{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)}\{\log p_{\boldsymbol{\theta}}(\mathbf{Y}_T)\} = &\int \log p_{\boldsymbol{\theta}}(\mathbf{X}_T, \mathbf{Y}_T)p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)d\mathbf{X}_T \\ &- \int \log p_{\boldsymbol{\theta}}(\mathbf{X}_T|\mathbf{Y}_T)p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)d\mathbf{X}_T \end{aligned} \tag{4.35}$$

As the left hand side of (4.35) is independent of $\mathbf{X}_T$, we can write

$$\begin{aligned} L(\boldsymbol{\theta}) &= \int \log p_{\boldsymbol{\theta}}(\mathbf{X}_T, \mathbf{Y}_T)p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)d\mathbf{X}_T - \int \log p_{\boldsymbol{\theta}}(\mathbf{X}_T|\mathbf{Y}_T)p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)d\mathbf{X}_T \\ &= \underbrace{E_{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)}\left[\log p_{\boldsymbol{\theta}}(\mathbf{X}_T, \mathbf{Y}_T)\right]}_{Q(\boldsymbol{\theta},\boldsymbol{\theta}')} - \underbrace{E_{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)}\left[\log p_{\boldsymbol{\theta}}(\mathbf{X}_T|\mathbf{Y}_T)\right]}_{V(\boldsymbol{\theta},\boldsymbol{\theta}')} \end{aligned} \tag{4.36}$$

Assume that the log-likelihood function L is evaluated at two consecutive parameter values $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. The estimation algorithm has to guarantee that the likelihood function of the current estimate $L(\boldsymbol{\theta})$ is larger than the likelihood function value of the previous $L(\boldsymbol{\theta}')$. In other words, we have to guarantee that the difference

$$L(\boldsymbol{\theta}) - L(\boldsymbol{\theta}') = \big(Q(\boldsymbol{\theta},\boldsymbol{\theta}') - Q(\boldsymbol{\theta}',\boldsymbol{\theta}')\big) + \big(V(\boldsymbol{\theta}',\boldsymbol{\theta}') - V(\boldsymbol{\theta},\boldsymbol{\theta}')\big) \tag{4.37}$$

is positive. Inserting (4.36) and only focusing on the second term of (4.37), we obtain

$$\begin{aligned} V(\boldsymbol{\theta}', \boldsymbol{\theta}') - V(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \int \log p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T) p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T) d\mathbf{X}_T \\ &\quad - \int \log p_{\boldsymbol{\theta}}(\mathbf{X}_T|\mathbf{Y}_T) p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T) d\mathbf{X}_T \\ &= \int \log \frac{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)}{p_{\boldsymbol{\theta}}(\mathbf{X}_T|\mathbf{Y}_T)} p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T) d\mathbf{X}_T \\ &= E_{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)} \left[-\log \frac{p_{\boldsymbol{\theta}}(\mathbf{X}_T|\mathbf{Y}_T)}{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)} \right] \end{aligned} \tag{4.38}$$

which is the Kullback-Leibler information and is used to describe the measure of agreement between two probability density functions (Kárný, 1996; Kullback and Leibler, 1951). Noting that negative logarithm is a convex function and applying Jensen's inequality (see A.4), which states that

$$E\left[f(x)\right] \geq f\left(E\right]x]) \tag{4.39}$$

we can derive the following expression

$$\begin{aligned} E_{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)} \left[-\log \frac{p_{\boldsymbol{\theta}}(\mathbf{X}_T|\mathbf{Y}_T)}{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)} \right] &\geq -\log E_{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)} \left[\frac{p_{\boldsymbol{\theta}}(\mathbf{X}_T|\mathbf{Y}_T)}{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)} \right] \\ &= -\log \int \left[\frac{p_{\boldsymbol{\theta}}(\mathbf{X}_T|\mathbf{Y}_T)}{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)} \right] p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T) d\mathbf{X}_T \\ &= -\log \int p_{\boldsymbol{\theta}}(\mathbf{X}_T|\mathbf{Y}_T) d\mathbf{X}_T = 0 \end{aligned} \tag{4.40}$$

which shows that $V(\boldsymbol{\theta}', \boldsymbol{\theta}') - V(\boldsymbol{\theta}, \boldsymbol{\theta}') \geq 0$. Therefore, any value of the parameter vector $\boldsymbol{\theta}$, that will guarantee $Q(\boldsymbol{\theta}, \boldsymbol{\theta}') - Q(\boldsymbol{\theta}', \boldsymbol{\theta}') \geq 0$, will result in $L(\boldsymbol{\theta}) - L(\boldsymbol{\theta}') \geq 0$. In other words, if we select a new parameter vector value $\boldsymbol{\theta}$ in a way that it will increase the value of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$, this value will indeed increase the value of $L(\boldsymbol{\theta})$. The optimal performance is achieved, if the selection of $\boldsymbol{\theta}$ is done by some optimization procedure.

The Expectation Maximization algorithm

1. Start with initial parameter estimate $\boldsymbol{\theta}_0$ and $k = 0$.

2. **Expectation step (E-step)**:
   Compute the expected value of the complete data log-likelihood function.

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = E_{p_{\boldsymbol{\theta}_k}(\mathbf{X}_T|\mathbf{Y}_T)} \left[\log p_{\boldsymbol{\theta}}(\mathbf{X}_T, \mathbf{Y}_T) \right] \tag{4.41}$$

3. **Maximization step (M-step)**:
   Compute the optimal parameter vector value by maximizing the function $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$.

$$\boldsymbol{\theta}_{k+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k) \tag{4.42}$$

4. If convergence criteria are not satisfied, set $k = k + 1$ and return to step 2.

One iteration of the EM algorithm is shown in Figure 4.1, where we can see, that finding an estimate for $\boldsymbol{\theta}$ that will increase $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$ will in turn also increase $L(\boldsymbol{\theta})$.

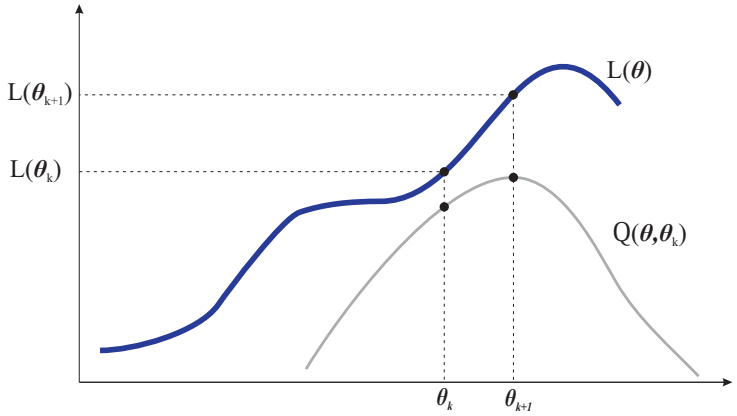


Figure 4.1: One iteration of the EM algorithm.

4.4 The EM algorithm for linear system identification

We will first address a special case when the model is linear and the noise distributions are Gaussian. The derivation adopts ideas from Shumway and Stoffer (2005). Assuming that a linear state-space model is given in the following form

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t \tag{4.43a}$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t + \mathbf{v}_t \tag{4.43b}$$

where $\mathbf{y}_t$ is the observed output, $\mathbf{u}_t$ is the known input, $\mathbf{x}_t$ is system state, $\mathbf{w}_t$ and $\mathbf{v}_t$ are zero mean i.i.d. vector processes with a positive-semidefinite covariance matrix and the following Gaussian distribution

$$\begin{bmatrix} \mathbf{w}_t \\ \mathbf{v}_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{Q}, & \mathbf{S} \\ \mathbf{S}^T, & \mathbf{R} \end{bmatrix} \right) \tag{4.44}$$

The distribution of the initial state is also Gaussian, given as

$$\mathbf{x}_1 \sim \mathcal{N}(\boldsymbol{\mu_1}, \mathbf{P}_1) \tag{4.45}$$

According to the Algorithm 4.1, the first task is to compute the expected value of the complete data log-likelihood function

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = E_{p_{\boldsymbol{\theta}_k}(\mathbf{X}_T|\mathbf{Y}_T)} \left[\log p_{\boldsymbol{\theta}}(\mathbf{X}_T, \mathbf{Y}_T)\right] \tag{4.46}$$

where the joint likelihood of the measured output and system states can be written as

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{y}_1, \ldots, \mathbf{y}_T, \mathbf{x}_1, \ldots, \mathbf{x}_T) &= p_{\boldsymbol{\theta}}(\mathbf{y}_1, \ldots, \mathbf{y}_T | \mathbf{x}_1, \ldots, \mathbf{x}_T) p_{\boldsymbol{\theta}}(\mathbf{x}_1, \ldots, \mathbf{x}_T) \\ &= p_{\boldsymbol{\theta}}(\mathbf{x}_1) \prod_{t=1}^{T-1} p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{x}_t) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t) \end{aligned} \tag{4.47}$$

Taking into account Gaussian distributions and ignoring the constants, the complete data likelihood function can be written as

$$\begin{aligned} -2\log p_{\boldsymbol{\theta}}(\mathbf{X}_T, \mathbf{Y}_T) = &\log|\mathbf{P}_1| + (\mathbf{x_1} - \boldsymbol{\mu}_1)^T \mathbf{P}_1^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1) \\ &+ T\log|\mathbf{Q}| + \sum_{t=1}^{T} (\mathbf{x}_{t+1} - \mathbf{A}\mathbf{x}_t - \mathbf{B}\mathbf{u}_t)^T \mathbf{Q}^{-1} (\mathbf{x}_{t+1} - \mathbf{A}\mathbf{x}_t - \mathbf{B}\mathbf{u}_t) \\ &+ T\log|\mathbf{R}| + \sum_{t=1}^{T} (\mathbf{y}_t - \mathbf{C}\mathbf{x}_t - \mathbf{D}\mathbf{u}_t)^T \mathbf{R}^{-1} (\mathbf{y}_t - \mathbf{C}\mathbf{x}_t - \mathbf{D}\mathbf{u}_t) \end{aligned} \tag{4.48}$$

where $\mathbf{x}_1 \sim N(\boldsymbol{\mu}_1, \mathbf{P}_1)$. Taking the expected value of the expression with respect to the current parameter estimate $\boldsymbol{\theta}_k$ and complete observed data $\mathbf{Y}_T$ leads to (c.f. (4.46))

$$\begin{aligned} -2Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = &\log|\mathbf{P}_1| + \mathrm{tr}\left[\mathbf{P}_1^{-1} E\left[(\mathbf{x}_1 - \boldsymbol{\mu}_1)(\mathbf{x}_1 - \boldsymbol{\mu}_1)^T\right]\right] \\ &+ T\log|\mathbf{Q}| + \mathrm{tr}\left[\mathbf{Q}^{-1} E\left[\sum_{t=1}^{T} \left((\mathbf{x}_{t+1} - \mathbf{A}\mathbf{x}_t - \mathbf{B}\mathbf{u}_t)(\mathbf{x}_{t+1} - \mathbf{A}\mathbf{x}_t - \mathbf{B}\mathbf{u}_t)^T\right)\right]\right] \\ &+ T\log|\mathbf{R}| + \mathrm{tr}\left[\mathbf{R}^{-1} E\left[\sum_{t=1}^{T} \left((\mathbf{y}_t - \mathbf{C}\mathbf{x}_t - \mathbf{D}\mathbf{u}_t)(\mathbf{y}_t - \mathbf{C}\mathbf{x}_t - \mathbf{D}\mathbf{u}_t)^T\right)\right]\right] \end{aligned} \tag{4.49}$$

where we made use of the following relation

$$E\left[(\mathbf{x_1} - \boldsymbol{\mu}_1)^T \mathbf{P}_1^{-1} (\mathbf{x}_1 - \boldsymbol{\mu}_1)\right] = \mathrm{tr}\left[\mathbf{P}_1^{-1} E\left[(\mathbf{x}_1 - \boldsymbol{\mu}_1)(\mathbf{x}_1 - \boldsymbol{\mu}_1)^T\right]\right] \tag{4.50}$$

which holds if $E\left[(\mathbf{x}_1 - \boldsymbol{\mu}_1)\right] = 0$. Similar relations are also used for the second and the third row in Equation (4.49). To reduce complexity of the above equation, we introduce

the following notations

$$\boldsymbol{\Gamma} = \begin{bmatrix} \mathbf{A}, & \mathbf{B} \\ \mathbf{C}, & \mathbf{D} \end{bmatrix} \tag{4.51a}$$

$$\boldsymbol{\Pi} = \begin{bmatrix} \mathbf{Q}, & \mathbf{S} \\ \mathbf{S}^T, & \mathbf{R} \end{bmatrix} \tag{4.51b}$$

$$\widehat{\mathbf{x}}_{t|T} = E_{p_{\boldsymbol{\theta}_k}(\mathbf{X}_T|\mathbf{Y}_T)}\left[\mathbf{x}_t\right] \tag{4.51c}$$

$$\mathbf{z}_t^T = \left[\ \mathbf{x}_t,\ \mathbf{u}_t\ \right]^T \tag{4.51d}$$

$$\boldsymbol{\xi}_t^T = \left[\ \mathbf{x}_{t+1},\ \mathbf{y}_t\ \right]^T \tag{4.51e}$$

$$\boldsymbol{\Phi} = \frac{1}{T}\sum_{t=1}^{T} E_{p_{\boldsymbol{\theta}_k}(\mathbf{X}_T|\mathbf{Y}_T)}\left[\boldsymbol{\xi}_t\boldsymbol{\xi}_t^T|\mathbf{Y}_T\right] \tag{4.51f}$$

$$\boldsymbol{\Psi} = \frac{1}{T}\sum_{t=1}^{T} E_{p_{\boldsymbol{\theta}_k}(\mathbf{X}_T|\mathbf{Y}_T)}\left[\boldsymbol{\xi}_t\mathbf{z}_t^T|\mathbf{Y}_T\right] \tag{4.51g}$$

$$\boldsymbol{\Sigma} = \frac{1}{T}\sum_{t=1}^{T} E_{p_{\boldsymbol{\theta}_k}(\mathbf{X}_T|\mathbf{Y}_T)}\left[\mathbf{z}_t\mathbf{z}_t^T|\mathbf{Y}_T\right] \tag{4.51h}$$

The expectations required to compute (4.51) can be obtained using the Kalman smoother equations (Algorithm 3.2).

$$E_{p_{\boldsymbol{\theta}_k}(\mathbf{X}_T|\mathbf{Y}_T)}\left[\mathbf{x}_t\right] = \widehat{\mathbf{x}}_{t|T} \tag{4.52}$$

$$E_{p_{\boldsymbol{\theta}_k}(\mathbf{X}_T|\mathbf{Y}_T)}\left[\mathbf{x}_t\mathbf{x}_t^T\right] = \widehat{\mathbf{x}}_{t|T}\widehat{\mathbf{x}}_{t|T}^T + \mathbf{P}_{t|T} \tag{4.53}$$

$$E_{p_{\boldsymbol{\theta}_k}(\mathbf{X}_T|\mathbf{Y}_T)}\left[\mathbf{x}_{t+1}\mathbf{x}_t^T\right] = \widehat{\mathbf{x}}_{t+1|T}\widehat{\mathbf{x}}_{t|T}^T + \mathbf{M}_{t+1|T} \tag{4.54}$$

Using (4.51), the expression in (4.49) simplifies to (Gibson and Ninness, 2005)

$$\begin{aligned} -2Q(\boldsymbol{\theta},\boldsymbol{\theta}_k) = {} & \log|\mathbf{P}_1| + \operatorname{tr}\left\{\mathbf{P}_1(\widehat{\mathbf{x}}_{t|T}-\boldsymbol{\mu}_1)(\widehat{\mathbf{x}}_{t|T}-\boldsymbol{\mu}_1)^T + \mathbf{P}_1\right\} \\ & + T\log|\boldsymbol{\Pi}| + T\ \operatorname{tr}\left\{\boldsymbol{\Pi}^{-1}\left[\boldsymbol{\Phi} - \boldsymbol{\Psi}\boldsymbol{\Gamma}^T - \boldsymbol{\Gamma}\boldsymbol{\Psi}^T + \boldsymbol{\Gamma}\boldsymbol{\Sigma}\boldsymbol{\Gamma}^T\right]\right\} \end{aligned} \tag{4.55}$$

The task is to find the parameter values that will minimize the expression (4.55). First note, that the term

$$\operatorname{tr}\left\{\mathbf{P}_1(\widehat{\mathbf{x}}_{t|T}-\boldsymbol{\mu}_1)(\widehat{\mathbf{x}}_{t|T}-\boldsymbol{\mu}_1)^T + \mathbf{P}_1\right\}$$

is globally minimized with respect to $\boldsymbol{\mu}_1$ by

$$\boldsymbol{\mu}_1 = \widehat{\mathbf{x}}_{1|T} \tag{4.56}$$

Additionally, the term

$$\begin{aligned} & \operatorname{tr}\left\{\boldsymbol{\Pi}^{-1}\left[\boldsymbol{\Phi} - \boldsymbol{\Psi}\boldsymbol{\Gamma}^T - \boldsymbol{\Gamma}\boldsymbol{\Psi}^T + \boldsymbol{\Gamma}\boldsymbol{\Sigma}\boldsymbol{\Gamma}^T\right]\right\} = \\ & \operatorname{tr}\left\{\boldsymbol{\Pi}^{-1}\left[\left(\boldsymbol{\Gamma} - \boldsymbol{\Psi}\boldsymbol{\Sigma}^{-1}\right)\boldsymbol{\Sigma}\left(\boldsymbol{\Gamma} - \boldsymbol{\Psi}\boldsymbol{\Sigma}^{-1}\right)^T + \boldsymbol{\Phi} - \boldsymbol{\Psi}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Psi}^T\right]\right\} \end{aligned} \tag{4.57}$$

is minimized with respect to $\mathbf{\Gamma}$ by the following selection

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{A}, & \mathbf{B} \\ \mathbf{C}, & \mathbf{D} \end{bmatrix} = \mathbf{\Psi}\mathbf{\Sigma}^{-1} \tag{4.58}$$

The expression for $\mathbf{\Pi}$ can be derived by using Lemma 4.1 and the chain rule.

Lemma 4.1. *Suppose* $\mathbf{X}, \mathbf{Y} \in \mathbf{R}^{n\times n}$ *and* $\mathbf{X}$ *are invertible. Then (Magnus and Neudecker, 1999; Petersen and Pedersen, 2008)*

$$\frac{d}{d\mathbf{X}} \log |\mathbf{X}| = \mathbf{X}^{-1}, \qquad \frac{d}{d\mathbf{X}} \mathbf{X}^{-1} = -\mathbf{X}^{-2}, \qquad \frac{d}{d\mathbf{X}} \mathrm{tr}\left\{\mathbf{X}\mathbf{Y}\right\} = \mathbf{Y}^T \tag{4.59}$$

Taking the derivationof the second row in (4.55) with respect to $\mathbf{\Pi}$ gives

$$\frac{d}{d\mathbf{\Pi}} \log |\mathbf{\Pi}| + \frac{d}{d\mathbf{\Pi}} \mathrm{tr}\left\{\mathbf{\Pi}^{-1}\left[\mathbf{\Phi} - \mathbf{\Psi}\mathbf{\Gamma}^T - \mathbf{\Gamma}\mathbf{\Psi}^T + \mathbf{\Gamma}\mathbf{\Sigma}\mathbf{\Gamma}^T\right]\right\} = \mathbf{\Pi}^{-1} - \mathbf{\Pi}^{-1}\left(\mathbf{\Phi} - \mathbf{\Psi}\mathbf{\Sigma}\mathbf{\Psi}^T\right)\mathbf{\Pi}^{-1}$$

which is zero for the following choice

$$\mathbf{\Pi} = \begin{bmatrix} \mathbf{Q}, & \mathbf{S} \\ \mathbf{S}^T, & \mathbf{R} \end{bmatrix} = \mathbf{\Phi} - \mathbf{\Psi}\mathbf{\Sigma}\mathbf{\Psi}^T \tag{4.60}$$

The expression for $\mathbf{P}_1$ follows in the similar manner and the optimal value is

$$\mathbf{P}_1 = \mathbf{P}_{1|N} \tag{4.61}$$

Together, the expressions (4.56), (4.58), (4.60) and (4.61) describe a global maximizer of $O(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$ with respect to $\boldsymbol{\theta}$. We can now summarize the entire procedure in a relatively convenient and transparent algorithm.

Algorithm 4.7: The EM algorithm for linear system identification

1. Select suitable initial values for the model parameters, convergence criteria and set $k = 0$.

$$\boldsymbol{\theta}_0 = [\ \mathbf{A},\ \mathbf{B},\ \mathbf{C},\ \mathbf{D},\ \mathbf{Q},\ \mathbf{R}\ ] \tag{4.62}$$

2. **Expectation step (E-step)**: run the Kalman filter (Algorithm 3.1) and the Kalman smoother (Algorithm 3.2) and compute the quantities defined by (4.51).

3. **Maximization step (M-step)**: find maximum of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$ by setting the estimates of model parameters according to (4.56), (4.58), (4.60) and (4.61). These estimates of the parameter values are labeled $\boldsymbol{\theta}_{k+1}$.

4. If convergence criteria are not fulfilled, set $k = k + 1$ and return to step 2.

The algorithm is said to converge, when the difference in the data likelihood function between two successive steps falls below a predetermined threshold. The presented algorithm features closed form expressions and guarantees to find the optimal solution (in the ML sense) of the presented system identification problem. This is made possible due to the nature of linear transformation of random variable, which allows analytic solutions.

Example 4.3. Linear Gaussian system identification *Consider a linear dynamical model given by*

$$x_{t+1} = ax_t + w_t \tag{4.63a}$$

$$y_t = bx_t + v_t \tag{4.63b}$$

where w_t and v_t are i.i.d. zero mean random processes with variances q and r respectively. The parameter vector is defined as

$$\boldsymbol{\theta} = [a,\ b,\ q,\ r] = [\ 0.8,\ 0.4,\ 0.1,\ 0.1\ ] \tag{4.64}$$

Let us assume that the only unknown parameter is the parameter a and we wish to estimate it using the observed data $\mathbf{Y}_T = [y_1, y_2, \ldots, y_T]$, where $T = 100$. This can be done by a straightforward application of Algorithm 4.1, where we have set the maximum number of iterations to 20 and the initial parameter guess is $a_i = 0.5$.

```
Initial parameter value: a = 0.5
iteration 1, log-likelihood = -80.923712, parameter estimate 0.50
iteration 2, log-likelihood = -57.910966, parameter estimate 0.61
iteration 3, log-likelihood = -52.161095, parameter estimate 0.66
iteration 4, log-likelihood = -48.001501, parameter estimate 0.70
iteration 5, log-likelihood = -45.425422, parameter estimate 0.73
iteration 6, log-likelihood = -44.056047, parameter estimate 0.76
iteration 7, log-likelihood = -43.394530, parameter estimate 0.77
iteration 8, log-likelihood = -43.088465, parameter estimate 0.78
iteration 9, log-likelihood = -42.947547, parameter estimate 0.79
iteration 10, log-likelihood = -42.880907, parameter estimate 0.79
iteration 11, log-likelihood = -42.847399, parameter estimate 0.79
iteration 12, log-likelihood = -42.828790, parameter estimate 0.79
iteration 13, log-likelihood = -42.817066, parameter estimate 0.80
iteration 14, log-likelihood = -42.808705, parameter estimate 0.80
iteration 15, log-likelihood = -42.802147, parameter estimate 0.80
iteration 16, log-likelihood = -42.796681, parameter estimate 0.80
iteration 17, log-likelihood = -42.791971, parameter estimate 0.80
iteration 18, log-likelihood = -42.787843, parameter estimate 0.80
iteration 19, log-likelihood = -42.784195, parameter estimate 0.80
iteration 20, log-likelihood = -42.780959, parameter estimate 0.80
```

Listing 4.1: Matlab output of the EM algorithm for Example 4.3.

The algorithm found the optimal value after 13 iterations. Convergence of the optimization procedure is shown in Figure 4.2 and the entire optimization took 0.5 *seconds to compute.*

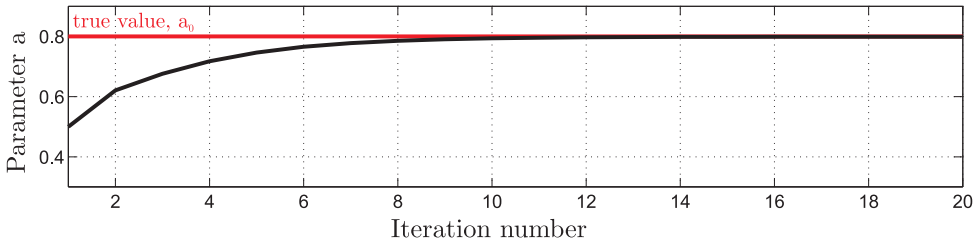


Figure 4.2: Iterations of the estimated parameter a from Example 4.3.

4.5 The UTEM: A novel unscented transformation EM algorithm for joint estimation of nonlinear systems

When trying to apply the expectation-maximization algorithm to a more general class of systems described by a nonlinear state-space model, the closed form solutions will disappear and we will have to rely on approximations. In order to keep computational load low, we will derive a computationally efficient approximation algorithm based on the unscented transform.

We start with the nonlinear state-space model

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\theta}, t) + \mathbf{w}_t \tag{4.65a}$$

$$\mathbf{y}_t = \mathbf{g}(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\theta}, t) + \mathbf{v}_t \tag{4.65b}$$

where $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are nonlinear functions parameterized by parameter vector $\boldsymbol{\theta}$. Additionally, $\mathbf{w}_t$ and $\mathbf{v}_t$ are process and measurement noise.

The first part of the algorithm is the *expectation* step (E-step), and requires calculation of the expected value of complete data log-likelihood function

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \int \log p_{\boldsymbol{\theta}}(\mathbf{X}_T, \mathbf{Y}_T) p_{\boldsymbol{\theta}'}(\mathbf{X}_T | \mathbf{Y}_T) d\mathbf{X}_T \\ &= E_{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)} \left[\log p_{\boldsymbol{\theta}}(\mathbf{X}_T, \mathbf{Y}_T) \right] \end{aligned} \tag{4.66}$$

The problem can be divided into two subproblems. The first one is to find the probability density function $p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)$, where $\boldsymbol{\theta}'$ is a fixed value parameter vector. This problem directly corresponds to solving the well-known smoothing problem (Särkkä, 2008). The

second part concerns calculating the expected value of the function over the distribution of the system states.

4.5.1 The main idea: Implementing the UT to approximate the expected value of the function $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$

Given the distribution of the state vector with respect to a parameter estimate $\boldsymbol{\theta}'$, one has to calculate the expected value of expression (4.66). In nonlinear systems an analytical expression for the integral usually cannot be derived (Chitralekha *et al.*, 2009). Thus, most approaches rely either on the gradient based approximations or on the Monte Carlo techniques. The main idea of the Monte Carlo based methods can be presented by the following equation for a general nonlinear function $f(\cdot)$:

$$E\left[f(x)\right] = \int f(x)p(x)dx \approx \frac{1}{M}\sum_{i=1}^{M} f(x^i) \tag{4.67}$$

where x^i are i.i.d. samples form the probability distribution $p(x)$. In our case, this relates to sampling from the state probability density function $p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)$. Unfortunately, this type of sampling is very inefficient for higher dimensional problems and large data sets (Gopaluni, 2008), as the number of samples M needs to be relatively high. This seriously raises the computational load of the algorithm. To overcome this issue, we propose an approximation based on the unscented transformation.

In the EM algorithm, we have to approximate the expected value of the nonlinear function:

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\mathbf{X}_T, \mathbf{Y}_T) = \log p_{\boldsymbol{\theta}}(\mathbf{x_1}) + \sum_{t=1}^{T-1} \log p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{x}_t) \\ + \sum_{t=1}^{T} \log p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t),\end{aligned} \tag{4.68}$$

The function for the E-step of the EM procedure $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ (4.66) can be viewed as the expected value of logarithm of the posterior probability density function $p_{\boldsymbol{\theta}}(\mathbf{X}_T, \mathbf{Y}_T)$.

Taking expectations of expression 4.68 yields

$$\begin{aligned} Q(\boldsymbol{\theta},\boldsymbol{\theta}') &= E_{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)}\left[\log p_{\boldsymbol{\theta}}(\mathbf{X}_T,\mathbf{Y}_T)\right] \\ &= \int \log p_{\boldsymbol{\theta}}(\mathbf{x}_1) p_{\boldsymbol{\theta}'}(\mathbf{x}_1|\mathbf{Y}_T) d\mathbf{x}_1 \\ &+ \sum_{t=1}^{T-1} \int\int \log p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{x}_t) p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1},\mathbf{x}_t|\mathbf{Y}_T) d\mathbf{x}_{t+1} d\mathbf{x}_t \\ &+ \sum_{t=1}^{T} \int \log p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t) p_{\boldsymbol{\theta}'}(\mathbf{x}_t|\mathbf{Y}_T) d\mathbf{x}_t \end{aligned} \tag{4.69}$$

To approximate the value of $Q(\boldsymbol{\theta},\boldsymbol{\theta}')$ for some prior parameter estimate $\boldsymbol{\theta}'$ we select a set of points $\boldsymbol{\mathcal{X}}_t^i$, along with the corresponding weights w_t^i based on the probability density functions $p_{\boldsymbol{\theta}'}(\mathbf{x}_1|\mathbf{Y}_T)$, $p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1},\mathbf{x}_t|\mathbf{Y}_T)$ and $p_{\boldsymbol{\theta}'}(\mathbf{x}_t|\mathbf{Y}_T)$.

Equation (4.69) can then be approximated using the unscented transformation based on Equation (3.42) as

$$\begin{aligned} Q(\boldsymbol{\theta},\boldsymbol{\theta}') &\approx \sum_{i=0}^{2N} w_1^i \log p_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{X}}_1^i) \\ &+ \sum_{t=1}^{T-1}\sum_{j=0}^{4N} w_t^j \log p_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{X}}_t^j|\boldsymbol{\mathcal{X}}_{t-1}^j) \\ &+ \sum_{t=1}^{T}\sum_{i=0}^{2N} w_t^i \log p_{\boldsymbol{\theta}}(\mathbf{y}_t|\boldsymbol{\mathcal{X}}_t^i) \end{aligned} \tag{4.70}$$

Where $\boldsymbol{\mathcal{X}}_1^i$ are sigma points representing the distribution $p_{\boldsymbol{\theta}'}(\mathbf{x}_1|\mathbf{Y}_T)$, similarly $\left[\boldsymbol{\mathcal{X}}_{t+1}^j, \boldsymbol{\mathcal{X}}_t^j\right]$ represents the joint distribution $p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1},\mathbf{x}_t|\mathbf{Y}_T)$ and $\boldsymbol{\mathcal{X}}_t^i$ represents $p_{\boldsymbol{\theta}'}(\mathbf{x}_t|\mathbf{Y}_T)$. The resulting expression (4.70) is an approximation of the function $Q(\boldsymbol{\theta},\boldsymbol{\theta}')$ (4.69).

The number or calculations required for approximating $Q(\boldsymbol{\theta},\boldsymbol{\theta}')$ depends only on the dimension of the state vector N. This means that at every time step t only a fixed number of computations is required. The computational complexity of the step is thus $\mathcal{O}(4N+1)$. Compared to the computational complexity of (4.67), which is $\mathcal{O}(M)$, where M is the number of samples used and $M \propto N^2$, this represents a significant decrease of complexity, especially as the dimension of the problem N increases.

Algorithm 4.8: The UTEM algorithm for nonlinear system identification

1. Select the initial values for model parameters $\boldsymbol{\theta}_0$, convergence criteria and set $k = 0$..
2. **Expectation step (E-step)**: run the unscented Kalman filter (Algorithm 3.4) and the unscented Kalman smoother (Algorithm 3.5) and store the estimated state vector $\mathbf{x}_{t|T}$ and the state covariance $\mathbf{P}_{t|T}$, $t = 1,\ldots,T$.

3. Define *sigma points* that approximate the distributions of the system states $\mathfrak{X}_1^i$ and the joint distribution $\left[\mathfrak{X}_{t+1}^j, \mathfrak{X}_t^j\right]$ according to (3.46).

4. **Maximization step (M-step)**: find the maximum of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$ using the sigma points and the equation (4.70) to approximate the expected value. The parameter values provide an improved estimate and are labeled $\boldsymbol{\theta}_{k+1}$.

5. If the convergence criteria are not fulfilled, set $k = k + 1$ and return to step 2.

Example 4.4. Linear Gaussian system identification *Consider a linear dynamical model given by*

$$x_{t+1} = ax_t + w_t \tag{4.71a}$$

$$y_t = bx_t + v_t \tag{4.71b}$$

where w_t and v_t are i.i.d. zero mean random processes with variances q and r respectively. The parameter vector is defined as

$$\boldsymbol{\theta} = [\ a,\ b,\ q,\ r\ ] = [\ 0.8,\ 0.4,\ 0.1,\ 0.1\ ] \tag{4.72}$$

Let us assume, that the only unknown parameter is the parameter a and we wish to estimate it using the observed data $\mathbf{Y}_T = [y_1, y_2, \ldots, y_T]$, where $T = 100$. This can be done by a straightforward application of Algorithm 4.2, where we have set the maximum number of iterations to 100 and initial parameter guess is $a_i = 0.5$.

```
Initial parameter value: a = 0.5
Iteration 2, Q-function value: 186.47, parameter estimate   0.61
Iteration 3, Q-function value: 167.53, parameter estimate   0.70
Iteration 4, Q-function value: 158.80, parameter estimate   0.75
Iteration 5, Q-function value: 157.06, parameter estimate   0.78
Iteration 6, Q-function value: 157.61, parameter estimate   0.79
Iteration 7, Q-function value: 158.14, parameter estimate   0.80
Iteration 8, Q-function value: 158.40, parameter estimate   0.80
Iteration 9, Q-function value: 158.40, parameter estimate   0.80
Iteration 10, Q-function value: 158.40, parameter estimate   0.80
Iteration 11, Q-function value: 158.40, parameter estimate   0.80
Iteration 12, Q-function value: 158.40, parameter estimate   0.80
Iteration 13, Q-function value: 158.40, parameter estimate   0.80
Iteration 14, Q-function value: 158.40, parameter estimate   0.80
Iteration 15, Q-function value: 158.40, parameter estimate   0.80
Iteration 16, Q-function value: 158.40, parameter estimate   0.80
Iteration 17, Q-function value: 158.40, parameter estimate   0.80
Iteration 18, Q-function value: 158.40, parameter estimate   0.80
Iteration 19, Q-function value: 158.40, parameter estimate   0.80
Iteration 20, Q-function value: 158.40, parameter estimate   0.80
```

Listing 4.2: Matlab output of the EM algorithm for Example 4.4.

The algorithm found the optimal value after 7 iterations. Graphical representation of the optimization procedure is shown in Figure 4.3 and the entire optimization took approximately 4 seconds to compute.

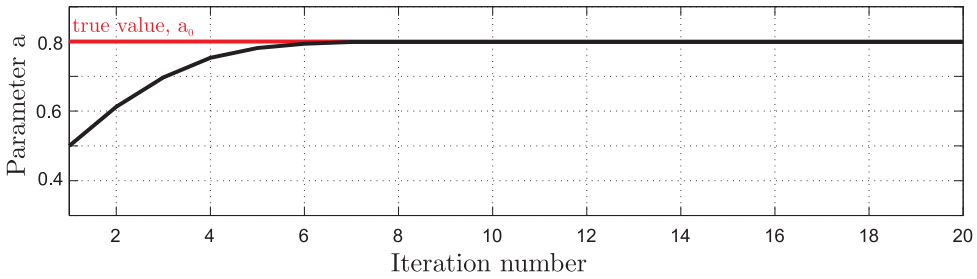


Figure 4.3: Iterations of the estimated parameter a from Example 4.4.

4.6 Sequential Monte Carlo EM algorithm for joint estimation of nonlinear systems

The EM algorithm based on unscented transformation enables computationally undemanding calculation of the ML estimates of the model parameters in nonlinear systems. However, this comes at the cost of approximation errors. To address the problem of joint estimation in nonlinear and non-Gaussian systems, more accurate approximation methods have to be used.

The problem has recently been solved by Schön *et al.* (2011) and in what follows we will apply their ideas. As in the previous algorithm, we focus on approximation of the expected value of the joint data log likelihood function.

$$\begin{aligned} Q(\boldsymbol{\theta},\boldsymbol{\theta}') &= E_{p_{\boldsymbol{\theta}'}(\mathbf{X}_T|\mathbf{Y}_T)}\left[\log p_{\boldsymbol{\theta}}(\mathbf{X}_T,\mathbf{Y}_T)\right] \\ &= \int \log p_{\boldsymbol{\theta}}(\mathbf{x}_1) p_{\boldsymbol{\theta}'}(\mathbf{x}_1|\mathbf{Y}_T) d\mathbf{x}_1 \\ &+ \sum_{t=1}^{T-1} \int\int \log p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{x}_t) p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1},\mathbf{x}_t|\mathbf{Y}_T) d\mathbf{x}_{t+1} d\mathbf{x}_t \\ &+ \sum_{t=1}^{T} \int \log p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t) p_{\boldsymbol{\theta}'}(\mathbf{x}_t|\mathbf{Y}_T) d\mathbf{x}_t \end{aligned} \tag{4.73}$$

Using results from the particle filter and the particle smoother (Algorithm 3.5 and Algorithm 3.6), the approximations of the first and the last integral in (4.73) is straightforward

and can be written as

$$\int \log p_{\boldsymbol{\theta}}(\mathbf{x}_1) p_{\boldsymbol{\theta}'}(\mathbf{x}_1|\mathbf{Y}_T) d\mathbf{x}_1 \approx \sum_{i=1}^{N} \gamma(\mathbf{x}_{1|T}^{(i)}) \log p_{\boldsymbol{\theta}}(\mathbf{x}_{1|T}^{(i)}) \tag{4.74}$$

$$\sum_{t=1}^{T} \int \log p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t) p_{\boldsymbol{\theta}'}(\mathbf{x}_t|\mathbf{Y}_T) d\mathbf{x}_t \approx \sum_{t=1}^{T} \sum_{i=1}^{N} \gamma(\mathbf{x}_{t|T}^{(i)}) \log p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_{t|T}^{(i)}) \tag{4.75}$$

where particle weights $\gamma(\mathbf{x}_{t|T}^{(i)})$ are computed by Algorithm 3.5 and Algorithm 3.6, using the known model structure and the parameter estimates $\boldsymbol{\theta}'$. The approximation of the second integral is less straightforward. The second integral in (4.73) depends on the joint probability density function $p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{y}_T)$ and using the particle filtering framework it can be approximated as

$$\sum_{t=1}^{T-1} \int \int \log p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{x}_t) p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{Y}_T) d\mathbf{x}_{t+1} d\mathbf{x}_t \approx \sum_{t=1}^{T-1} \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma(\mathbf{x}_{t|T}^{ij}) \log p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(i)}) \tag{4.76}$$

where the weights are computed by

$$\gamma(\mathbf{x}_{t|T}^{ij}) = \frac{\gamma(\mathbf{x}_{t|t-1}^{(i)}) \gamma(\mathbf{x}_{t+1|T}^{(j)}) p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}^{(j)}|\mathbf{x}_t^{(i)})}{\sum_{l=1}^{N} \gamma(\mathbf{x}_{t|t-1}^{(l)}) p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}^{(l)}|\mathbf{x}_t^{(l)})} \tag{4.77}$$

The result derives in the following way. Using Bayes' rule, we can write

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{Y}_T) = p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_T) p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{Y}_T) \tag{4.78}$$

Using the Markov property of the system states, the first term simplifies to

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_T) &= p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_t, \mathbf{Y}_{t+1:T}) \\ &= \frac{p_{\boldsymbol{\theta}}(\mathbf{Y}_{t+1:T}|\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{Y}_t) p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_t)}{p_{\boldsymbol{\theta}}(\mathbf{Y}_{t+1:T}|\mathbf{x}_{t+1}, \mathbf{Y}_t)} \\ &= p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_t) \end{aligned} \tag{4.79}$$

This result can be explained by the fact that if we know the value of $\mathbf{x}_{t+1}$, knowledge of $\mathbf{x}_t$ does not provide any additional information about $\mathbf{Y}_{t+1|T}$. Therefore

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{Y}_T) &= p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_t) p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{Y}_T) \\ &= \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{x}_t) p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{Y}_t)}{p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{Y}_t)} p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{Y}_T) \end{aligned} \tag{4.80}$$

Using this result, the integral in (4.76) can be rewritten as

$$\begin{aligned}
&\int\int \log p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{x}_t) p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{Y}_T) d\mathbf{x}_{t+1} d\mathbf{x}_t \\
&= \int \frac{p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}|\mathbf{Y}_T)}{p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}|\mathbf{Y}_t)} \left[\int \log p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{x}_t) p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}|\mathbf{x}_t) p_{\boldsymbol{\theta}'}(\mathbf{x}_t|\mathbf{Y}_t) d\mathbf{x}_t \right] d\mathbf{x}_{t+1} \\
&\approx \sum_{i=1}^{M} \gamma(\mathbf{x}_{t|t-1}^{(i)}) \int \frac{p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}|\mathbf{Y}_T)}{p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}|\mathbf{Y}_t)} \log p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) d\mathbf{x}_{t+1} \\
&\approx \sum_{i=1}^{M} \sum_{j=1}^{M} \gamma(\mathbf{x}_{t|t-1}^{(i)}) \gamma(\mathbf{x}_{t+1|T}^{(j)}) \frac{p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}^{(j)}|\mathbf{x}_t^{(i)})}{p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}^{(j)}|\mathbf{Y}_t)} \log p_{\boldsymbol{\theta}}(\mathbf{x}_{t+1}^{(j)}|\mathbf{x}_t^{(i)})
\end{aligned} \tag{4.81}$$

Finally, the denominator is given by the particle filter and the application of the law of total probability.

$$\begin{aligned}
p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}^{(j)}|\mathbf{Y}_t) &= \int p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}^{(j)}|\mathbf{x}_t) p_{\boldsymbol{\theta}'}(\mathbf{x}_t|\mathbf{Y}_t) d\mathbf{x}_t \\
&\approx \sum_{l=1}^{M} \gamma(\mathbf{x}_{t|t-1}^{(l)}) p_{\boldsymbol{\theta}'}(\mathbf{x}_{t+1}^{(j)}|\mathbf{x}_t^{(l)})
\end{aligned} \tag{4.82}$$

Algorithm 4.9: The SMCEM algorithm for nonlinear system identification

1. Select the initial values for model parameters $\boldsymbol{\theta}_0$, convergence criteria and set $k = 0$.

2. **Expectation step (E-step)**: run the particle filter (Algorithm 3.5) and the particle smoother (Algorithm 3.6) and store the particles $\mathbf{x}_{t|T}^{(i)}$ and their corresponding weights $\gamma(\mathbf{x}_{t|T}^{(i)})$.

3. Use the result to form an approximation to $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$ according to (4.74), (4.76) and (4.75).

4. **Maximization step (M-step)**: find the maximum of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$. The parameter values provide an improved estimate and are labeled $\boldsymbol{\theta}_{k+1}$.

5. If convergence criteria are not fulfilled, set $k = k + 1$ and return to step 2.

Most of the attention here was focused on the expectation (E-step) of the algorithm and it is crucial to note that the maximization (M-step) is equally important. Any standard nonlinear programming technique can be applied. In our case, we used the standard nonlinear optimization algorithm, implemented in MATLAB function `fminsearch`.

Example 4.5. Linear Gaussian system identification
Consider a linear dynamical model given by

$$x_{t+1} = ax_t + w_t \tag{4.83a}$$

$$y_t = bx_t + v_t \tag{4.83b}$$

where w_t and v_t are i.i.d. zero mean random processes with variances q and r, respectively. The parameter vector is defined as

$$\boldsymbol{\theta} = [\ a,\ b,\ q,\ r\ ] = [\ 0.8,\ 0.4,\ 0.1,\ 0.1\ ] \tag{4.84}$$

We take a as the only unknown parameter and we wish to estimate it using the observed data $\mathbf{Y}_T = [y_1, y_2, \ldots, y_T]$, where $T = 100$. This can be done by application of Algorithm 4.3, where we have set the maximum number of iterations to 20, $M = 100$ particles were used and the initial parameter guess is $a_i = 0.5$.

```
Initial parameter guess: 0.5
Iteration 1, Q-function value: 135.91, parameter estimate   0.64
Iteration 2, Q-function value: 132.40, parameter estimate   0.74
Iteration 3, Q-function value: 131.78, parameter estimate   0.78
Iteration 4, Q-function value: 132.05, parameter estimate   0.80
Iteration 5, Q-function value: 129.84, parameter estimate   0.79
Iteration 6, Q-function value: 130.50, parameter estimate   0.80
Iteration 7, Q-function value: 131.89, parameter estimate   0.80
Iteration 8, Q-function value: 131.28, parameter estimate   0.79
Iteration 9, Q-function value: 131.56, parameter estimate   0.80
Iteration 10, Q-function value: 130.81, parameter estimate   0.80
Iteration 11, Q-function value: 130.05, parameter estimate   0.80
Iteration 12, Q-function value: 131.89, parameter estimate   0.81
Iteration 13, Q-function value: 130.03, parameter estimate   0.81
Iteration 14, Q-function value: 130.85, parameter estimate   0.80
Iteration 15, Q-function value: 129.78, parameter estimate   0.80
Iteration 16, Q-function value: 129.41, parameter estimate   0.81
Iteration 17, Q-function value: 128.71, parameter estimate   0.80
Iteration 18, Q-function value: 130.40, parameter estimate   0.80
Iteration 19, Q-function value: 129.81, parameter estimate   0.80
Iteration 20, Q-function value: 129.73, parameter estimate   0.80
```

Listing 4.3: Matlab output of the EM algorithm for Example 4.5.

The algorithm found the optimal value after only 4 iterations. We can observe that the parameter estimate does not settle to a fixed value due to the Monte-Carlo nature of the algorithm. Graphical representation of the optimization procedure is shown in Figure 4.4 and the enture optimization took almost 9 seconds to compute.

4.7 Computational complexity of the algorithms

The main advantage of the UTEM algorithm lies in its low computational complexity compared to the Monte Carlo methods. The empirical evidence for this can be seen by

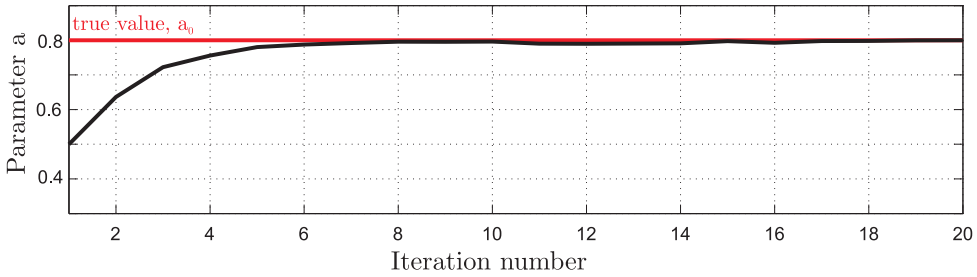


Figure 4.4: Iterations of the estimated a from Example 4.5.

comparing the computational times in Example 4.4 (approximately 4 seconds) and Example 4.5 (almost 9 seconds). However, to fully understand this relation, we will analyze the computational complexity of both algorithms for nonlinear system identification.

The particle smoother requires $\mathcal{O}(N \cdot M)$ computations for each point in the data set, where N is the dimension of the state-space and M is the number of particles used. For the unscented Kalman smoother, this value is $\mathcal{O}((2N+1) \cdot N)$. Furthermore, the effective computation cost in particle smoother compared to the unscented Kalman smoother is higher, because it involves the calculation of exponential likelihood functions (Chitralekha *et al.*, 2009).

The difference in the computational load is even more evident in the maximization step. The number or calculations required for approximating $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ in the UTEM algorithm depends only on the dimension of the state vector N. This means that at every time step t only a fixed number of computations is required. The computational complexity of the step is thus $\mathcal{O}(4N+1)$. To form the expected value of the data log-likelihood function the SMCEM algorithm requires $\mathcal{O}(M)$ computations, where M is the number of samples used and $M \propto N^2$ function. The difference becomes more evident at high-dimensional state-spaces, because the number of particles M needed by the particle smoother grows exponentially.

4.8 Chapter summary

This chapter described an approach to the system identification of the state-space models using the maximum-likelihood estimation of unknown model parameters. The problem of maximizing the data likelihood function in the presence of hidden variables was solved by implementation of the expectation-maximization algorithm. Three different algorithms were presented, each aimed at a specific sub-class of models.

There closed form solution of the problem exists only for the linear and Gaussian state-space models. The expectation (E-step) of the algorithm is based on the Kalman filter and smoother algorithms and the maximization (M-step) can be solved analytically.

On the other hand, the nonlinear system identification algorithms have to rely on approximation methods. A novel numerically efficient algorithm based on unscented transformation is aimed at high dimensional problems, as its computational load can be kept at a relatively low level even when the state dimension increases. The idea of our algorithm concerns the use of unscented transformation to form the expected value of the log-likelihood function. Hence the computational complexity of the algorithm is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(2N+1)$, where N is the state vector dimension.

However, when faced with a highly nonlinear and non-Gaussian problems, more accurate approximations are needed. This was achieved by an implementation of the Monte Carlo methodology and resulted in the third algorithm that relies on the sequential Monte Carlo framework. Together, the three algorithms present a portfolio for solving the joint estimation problem in state-space models.

5 Performance analysis on the simulated examples

5.1 Nonlinear one-dimensional system

Performance of the unscented transformation EM algorithm (UTEM) and the sequential Monte Carlo EM algorithm (SMCEM) will be explored using a simulated example. The nonlinear state space model that will be used for this purpose was chosen because it is acknowledged as a challenging estimation problem (Arulampalam *et al.*, 2002; Doucet *et al.*, 2000; Schön *et al.*, 2011). The nonlinear model reads:

$$x_{t+1} = ax_t + b\frac{x_t}{1+x_t^2} + c\cos(1.2t) + w_t \tag{5.1a}$$

$$y_t = dx_t^2 + v_t \tag{5.1b}$$

$$w_t \sim \mathcal{N}(0, q) \tag{5.1c}$$

$$v_t \sim \mathcal{N}(0, r) \tag{5.1d}$$

$$\boldsymbol{\theta} = [\ a,\ b,\ c,\ d,\ q,\ r\ ] \tag{5.1e}$$

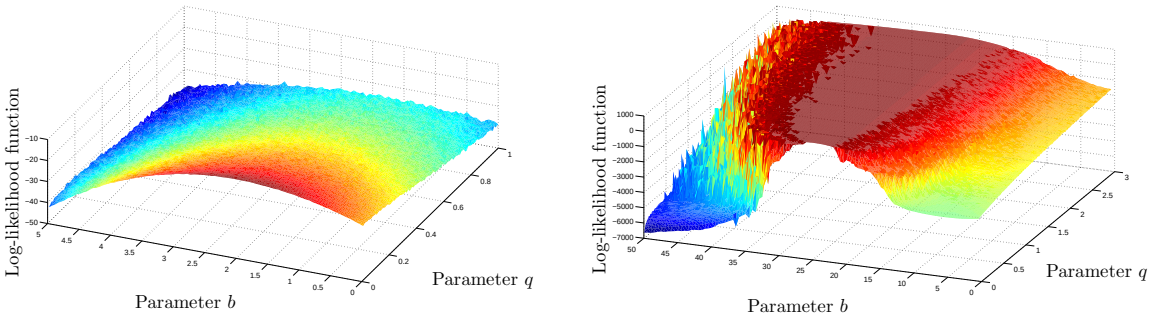
We will consider the parameter estimation problem for two different model parameter values, $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. Values of parameter vector $\boldsymbol{\theta}_1$ were selected to lie in a region without local optima in the log-likelihood function. The second set of model parameters $\boldsymbol{\theta}_2$ is adopted from the work of Doucet *et al.* (2000). The parameter values are given in Table 5.6.

Parameter	**Case I**: $\boldsymbol{\theta}_1$	**Case II**: $\boldsymbol{\theta}_2$
a	0.5	0.5
b	2	25
c	8	8
d	0.05	0.05
q	0	0
r	0.001	0.001

Table 5.1: Parameter values for parameter vectors $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$

A more detailed insight into the parameter estimation problem is obtained, if the log-likelihood function in the vicinity of a certain parameter vector is explored. This can be done in the following way. For a given parameter vector values, we compute the state estimates by the particle smoother algorithm using $M = 1000$ particles. Using the estimated state, we then compute the value of the log-likelihood function at the particular parameter values. By using a grid of parameter value, we get the plot of the log-likelihood function in a certain parameter subspace.

This procedure was adopted to determine the shape of the log-likelihood function for parameters b and q around the simulated (true) parameter values $\boldsymbol{\theta}_1$ (Figure 5.1(a)) and $\boldsymbol{\theta}_2$ (Figure 5.1(b)).



(a) $a = 0.5, c = 8, d = 0.05, r = 0.001$ (b) $a = 0.5, c = 8, d = 0.05, r = 0.001$

Figure 5.1: Log-likelihood function for model (5.1).

It can be seen that the log-likelihood function of these two parameters around the parameter vector $\boldsymbol{\theta}_1$ (Figure 5.1(a)) does not have any local maxima and therefore poses a less complex estimation problem. The two-dimensional log-likelihood function around the parameter vector $\boldsymbol{\theta}_2$ (Figure 5.1(b)) clearly exhibits irregular behavior with many local maxima. This is even more evident if the log-likelihood function is visualized as a function of b with $q = 0$ (Figure 5.2).

The algorithms for nonlinear system identification will be analyzed through a series of examples and Monte Carlo analyses in the following way.

- **The UTEM algorithm - Case I:** Two runs of the UTEM algorithm are presented, starting from different initial parameter guesses ($\boldsymbol{\theta}_{UT1}$ and $\boldsymbol{\theta}_{UT2}$).

- **The Monte Carlo analysis of the UTEM algorithm:** A Monte Carlo study using 100 different data realizations $\mathbf{Y}_T$ of length $T = 100$. For each realization the initial values of the estimated parameter vector $\boldsymbol{\theta}$ will be selected randomly from the uniform distribution defined in the interval equal to $\pm 50\%$ of the corresponding true parameter vector.

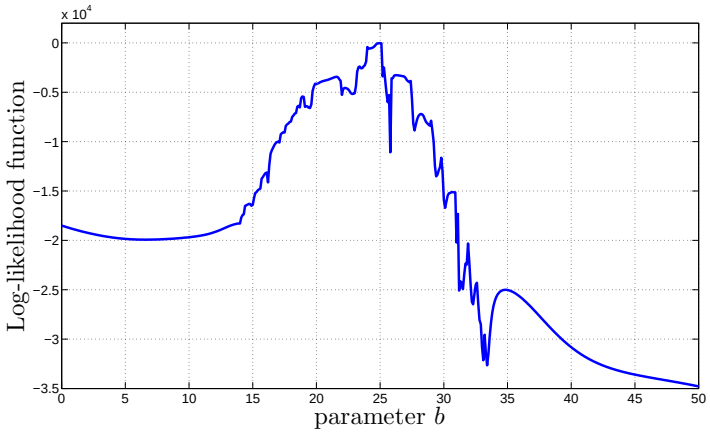


Figure 5.2: Log-likelihood function for model (5.1) at fixed parameter values $a = 0.5, b = 25, c = 8, d = 0.05, r = 0.001$.

- **The UTEM algorithm - Case II:** This will demonstrate that in certain cases the approximation errors become too large and preclude the algorithm from converging.
- **The SMCEM algorithm - Case II:** An estimation procedure of the SMCEM algorithm is used to explore the behavior of the algorithm. Specific attention is devoted to the robustness of the algorithm against finishing in local maxima.
- **The Monte Carlo analysis of the SMCEM algorithm:** The overall performance will be evaluated using a Monte Carlo analysis with 100 different data realizations $\mathbf{Y}_T$ of length $T = 100$. For each realization the initial values of the parameter vector $\boldsymbol{\theta}$ will be selected randomly from the uniform distribution defined in the interval equal to $\pm 50\%$ of the corresponding true parameter values.

5.1.1 Unscented transformation expectation maximization algorithm

The Algorithm 4.2 was implemented in MATLAB, where the maximization step was done using the Nelder-Mead optimization method implemented in function `fminsearch`. This section will present the results obtained on the simulated model data for two different true parameter values. However, while the implementation of the algorithm is relatively straightforward, the UT has three design parameters that have to be set beforehand.

Scaled unscented transformation design parameters

The scaled unscented transformation has three design parameters that determine location of the sigma points (κ, α, β). The value of β is not relevant for approximation of the

expected value of the expression (4.70), as it only determines the size of the covariance weights. In the unscented Kalman smoother and filter algorithms the value of β was set to the optimal choice for the Gaussian prior, which is $\beta = 2$. The value of the parameter κ should obey the condition $\kappa \geq 0$ to ensure positive semidefinitness of the covariance matrix. In the case when $N \leq 3$, it is possible to achieve the scaling invariance by setting $\kappa = 3 - N$ and thus canceling the effect of N on the sigma point placement.

The focus is turned to finding the optimal value of α. For this case we simulated the model output which consists of 100 time samples. The identification algorithm was repeated for different values of α. The algorithm started from the true parameter values and has been limited to the maximum of 50 iterations. The values of parameter α were set to the interval $0 < \alpha < 1$ with the increment of 0.01.

The resulting values of the function $Q(\theta, \theta')$ as a function of iteration number are shown in Figure 5.3.

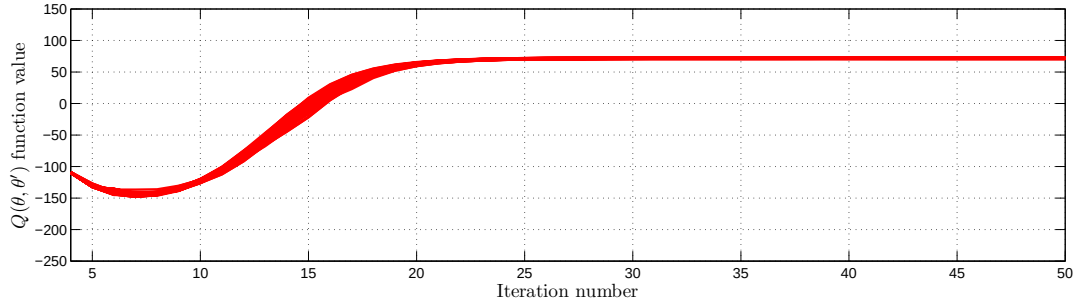


Figure 5.3: $Q(\theta, \theta')$ function values as a function of the iteration number for different values of the scaling parameter α.

The results show that the value of the scaling parameter α has very little or no effect on the quality of the final estimate. The result confirms the fact that the selection of the scaling parameter α is not crucial as long as it results in a numerically well-behaved set of sigma points and weights. Furthermore, the optimal values of all the parameters seem to be problem specific, which may be one of the reasons why relatively little research has been focused on determining the optimal settings of the UT design parameters. The algorithm was implemented with the unscented transformation parameter $\alpha = 0.7$.

The UTEM algorithm: Case I

For demonstration of the UTEM algorithm performance we consider the model given by 5.1, with the true parameter values set to $\boldsymbol{\theta}_1$. The output data were obtained by simulating the model and consist of $T = 100$ measured outputs. An excerpt from the estimation procedure is given below, showing the first ten iterations of the algorithm. The initial

parameters are set to

$$\boldsymbol{\theta}_{UT1} = [\ a,\ b,\ c,\ d,\ q,\ r\ ] = [\ 0.5,\ 3.0,\ 8.0,\ 0.05,\ 1,\ 0.01\ ] \tag{5.2}$$

The first 10 iterations from the algorithm run are given below.

```
Iteration: 1,
parameter estimate: [ 0.5 3.0 8.0 0.05 1 0.01 ]
Iteration: 2,
Q-function value: 211.66, parameter estimate: [ 0.449 3.218 7.838 0.051 0.845 0.011 ]
Iteration: 3,
Q-function value: 176.53, parameter estimate: [ 0.443 3.563 7.468 0.051 0.561 0.015 ]
Iteration: 4,
Q-function value: 147.74, parameter estimate: [ 0.481 2.747 7.838 0.052 0.508 0.017 ]
Iteration: 5,
Q-function value: 101.38, parameter estimate: [ 0.458 2.895 7.718 0.052 0.244 0.023 ]
Iteration: 6,
Q-function value:  78.89, parameter estimate: [ 0.491 2.256 7.749 0.052 0.233 0.025 ]
Iteration: 7,
Q-function value:  69.73, parameter estimate: [ 0.483 2.404 7.791 0.053 0.142 0.027 ]
Iteration: 8,
Q-function value:  56.87, parameter estimate: [ 0.484 2.294 7.743 0.053 0.148 0.027 ]
Iteration: 9,
Q-function value:  55.41, parameter estimate: [ 0.486 2.163 7.723 0.053 0.146 0.025 ]
Iteration: 10,
Q-function value:  47.57, parameter estimate: [ 0.497 2.155 7.703 0.054 0.116 0.024 ]
```

Listing 5.1: Output of the UTEM algorithm - Example 1a.

The final estimate for the model parameters for this run was

$$\widehat{\boldsymbol{\theta}}_{UT1} = [\ \widehat{a},\ \widehat{b},\ \widehat{c},\ \widehat{d},\ \widehat{q},\ \widehat{r}\ ] = [\ 0.498,\ 1.943,\ 7.570,\ 0.056,\ 0.049,\ 0.011\ ] \tag{5.3}$$

The trajectory of the UTEM algorithm estimates is again plotted in relation to the two dimensional log-likelihood function in Figure 5.4. The trajectory of the estimates projected in $q - b$ subspace takes irregular path towards the optimal values. This is due to the fact that the optimization problem has now six dimensions so that decrease of the log-likelihood function in a lower dimensional subspace does not necessarily imply decrease on the global level. Indeed, from the excerpt of the algorithm output it is clear that the Q-function values form a nondecreasing series and thus guaranteeing overall increase of the log-likelihood function.

The estimation was repeated by using the following initial parameter guesses:

$$\boldsymbol{\theta}_{UT2} = [\ a,\ b,\ c,\ d,\ q,\ r\ ] = [\ 1,\ 3.0,\ 8.0,\ 0.05,\ 0.0005,\ 0.01\ ] \tag{5.4}$$

This results in the following optimization procedure:

```
Iteration: 1,
parameter estimate: [ 1.000 3.000 8.000 0.050 0.0005 0.010 ]
```

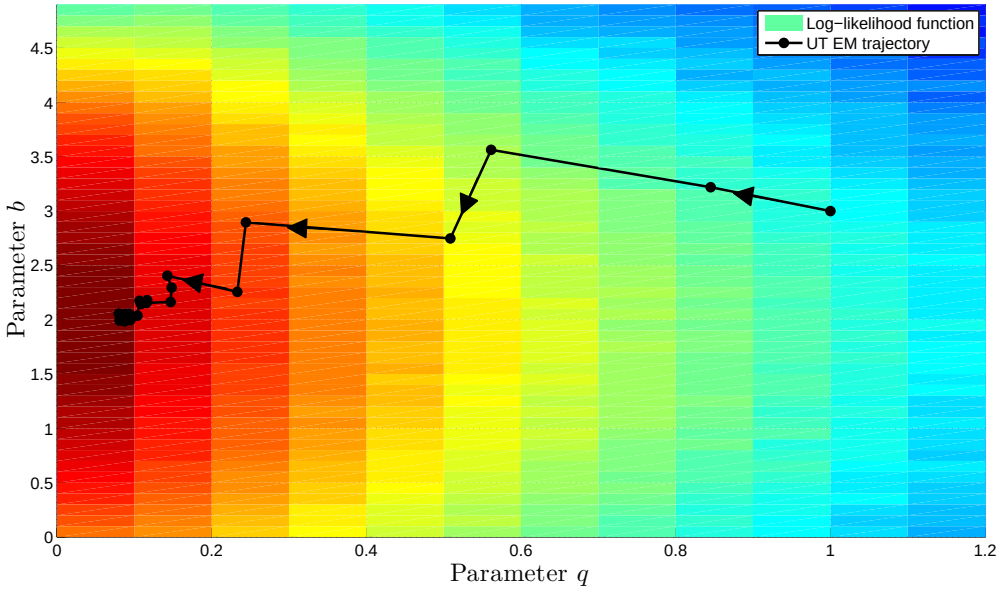


Figure 5.4: Trajectory of the parameter estimates from the UTEM algorithm - Example 1a projected on the data log-likelihood function in the $b - q$ parameter subspace.

```
Iteration: 2,
Q-function value: 5594.10, parameter estimate: [ 1.077 1.698 8.529 0.039 0.020 0.071 ]
Iteration: 3,
Q-function value: 581.70, parameter estimate: [ 1.062 1.167 8.565 0.044 0.019 0.303 ]
Iteration: 4,
Q-function value: 352.36, parameter estimate: [ 1.046 0.905 8.435 0.045 0.291 2.085 ]
Iteration: 5,
Q-function value: 343.80, parameter estimate: [ 1.017 -1.662 7.917 0.049 1.092 1.608 ]
Iteration: 6,
Q-function value: 375.74, parameter estimate: [ 0.880 -0.978 7.657 0.055 2.527 1.214 ]
Iteration: 7,
Q-function value: 374.18, parameter estimate: [ 0.729 -0.305 7.322 0.055 2.206 0.798 ]
Iteration: 8,
Q-function value: 349.02, parameter estimate: [ 0.653 -0.423 7.369 0.056 1.561 0.472 ]
Iteration: 9,
Q-function value: 309.65, parameter estimate: [ 0.618 -0.643 7.448 0.057 1.059 0.276 ]
Iteration: 10,
Q-function value: 263.52, parameter estimate: [ 0.587 -0.513 7.426 0.056 0.679 0.190 ]
```

Listing 5.2: Output of the UTEM algorithm - Example 1b.

After 100 iterations, the parameter estimates for this case are

$$\widehat{\boldsymbol{\theta}}_{UT3} = [\ \widehat{a},\ \widehat{b},\ \widehat{c},\ \widehat{d},\ \widehat{q},\ \widehat{r}\ ] = [\ 0.499,\ 1.823,\ 7.452,\ 0.057,\ 0.016,\ 0.006\ ] \tag{5.5}$$

The first 20 parameter estimates are plotted as a trajectory on two-dimensional log-likelihood surface (Figure 5.5).

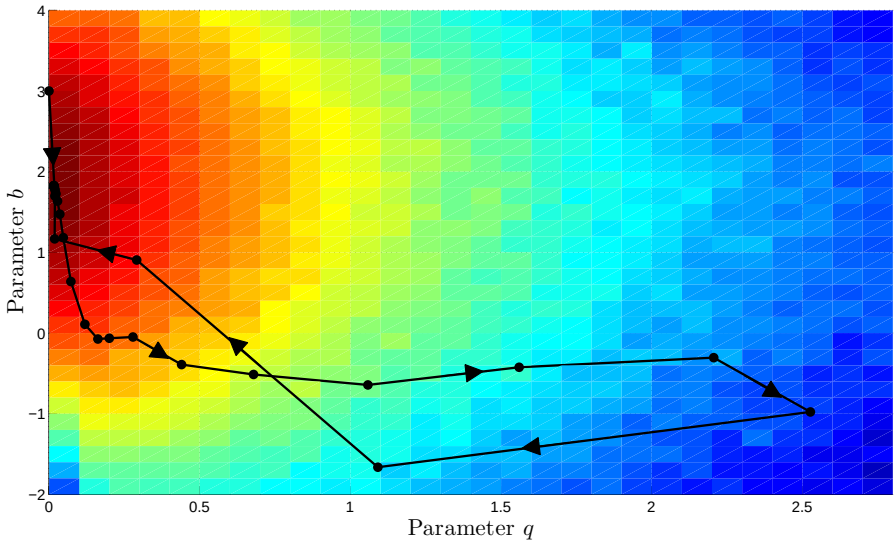


Figure 5.5: Trajectory of the parameter estimates from the UTEM algorithm - Example 1b projected on the data log-likelihood function in the $b - q$ parameter subspace.

Even though the estimation procedure started with the estimates of the parameters b and q relatively close to the true values, the trajectory shows complex behavior. There are two possible explanations for this. First one is that the approximation errors in this case are large and the convergence is slow, especially in the six-dimensional optimization problem. An alternative explanation may be that this type of behavior is associated with the EM algorithm, as similar observations were made by different authors (McLachlan and Krishnan, 2008; Schön *et al.*, 2011). However, this phenomena is not fully understood and to properly answer this question, further research is necessary.

The Monte Carlo analysis of the UTEM algorithm

In order to fully evaluate the effectiveness of the UTEM algorithm, a Monte Carlo study was performed. This was done by using 100 different data realizations of length $T = 100$. The estimate of model parameters was computed using 100 iterations of the UTEM algorithm. Prior to every run, the initial values of the parameter estimates were randomly selected from the uniform distribution on the interval corresponding to the $\pm 50\%$ of the true parameter value.

The resulting estimates of the parameters $\widehat{\boldsymbol{\theta}} = \left[\widehat{a},\ \widehat{b},\ \widehat{c},\ \widehat{d},\ \widehat{q},\ \widehat{r} \right]$ as a function of iteration number are shown in Figure 5.6 and the final sample mean and variance of the parameter estimates are given in Table 5.2.

The values of the $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ function as a function of iteration number are shown in Figure

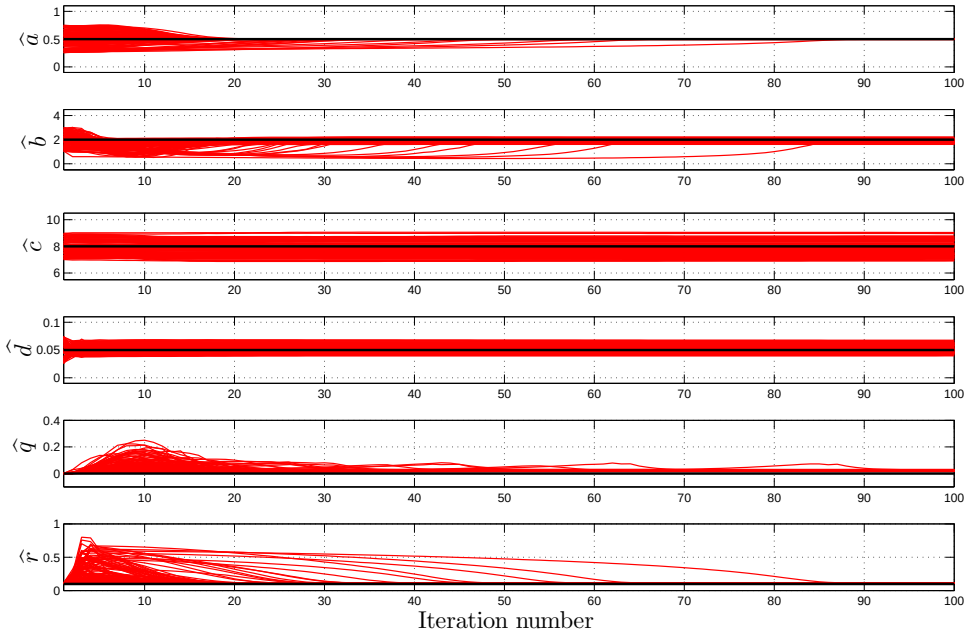


Figure 5.6: The parameter estimates as a function of the iteration number (black line represents the true parameter value).

Table 5.2: The statistics (mean and deviation) of the estimates obtained on a set of $N = 100$ Monte Carlo realizations of $\mathbf{Y}$.

Parameter	a	b	c	d	q	r
True	0.5	2	8	0.05	0	0.01
Estimated mean	0.503	2.02	8.35	0.046	0.03	0.006
Estimate variance	0.005	6.12	0.31	0.006	0.06	0.002

5.7. The presented results show that the algorithm converged in every run, independently of the initial parameter vector (within the selected interval) and system output realization.

The approximation error introduced by the UT approach can be visible in Figure 5.6. The final estimates of the parameter values differ from the true values. The magnitude of this error is partially connected to the characteristics of the nonlinearities and partially to the selection of the UT design parameters. Therefore, the quality of the estimate can be increased to some extent, if the UT design parameters were optimized with respect to the properties of the underlying dynamic model.

Finally, it is important to note that in this case, only 8 sigma points ($\mathcal{X}_i$) were used to estimate the statistics of the posterior distributions involved in the evaluation of the Equation 4.70. Because expression (4.70) has to be evaluated many times during the

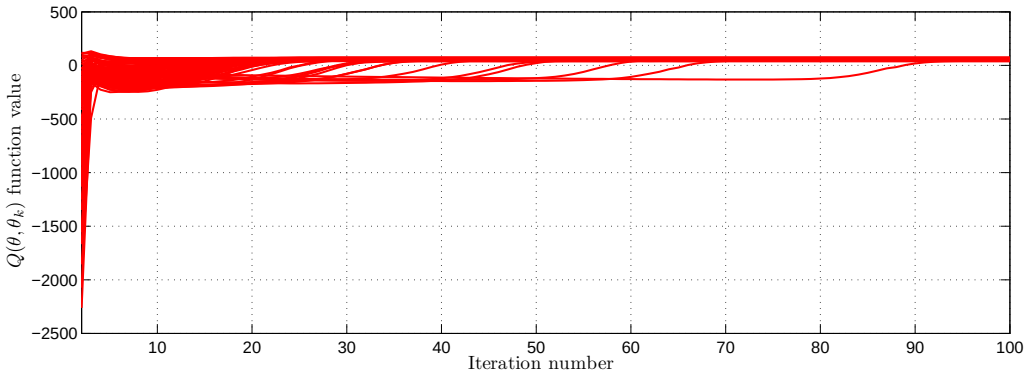


Figure 5.7: The $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ function values as a function of iteration number

maximization step, the UTEM algorithm has a relatively low computational load.

The UTEM algorithm - Case II

Consider the second set of model parameter values.

$$\boldsymbol{\theta}_{UT3} = [\ a,\ b,\ c,\ d,\ q,\ r\ ] = [\ 0.5,\ 25,\ 8,\ 0.05,\ 0,\ 0.001\ ] \tag{5.6}$$

Compared to previous parameter set, only parameter b is changed. The log-likelihood function shows irregular behavior, which can best be seen in Figure 5.2. This is connected to the highly nonlinear model dynamics in this region. The UTEM algorithm with the same design parameters as in previous example returns results as in Listing 5.3.

```
Iteration: 1  parameter estimates:  [ b = 10  q = 0.1 ]
Iteration: 2  Q-function value: 165454.48 parameter estimates: [ 0.4367  91.8975 ]
Iteration: 3  Q-function value: 171210.50 parameter estimates: [ 3.1365  76.9352 ]
Iteration: 4  Q-function value: 153504.94 parameter estimates: [ 3.2076  67.4125 ]
Iteration: 5  Q-function value: 139102.87 parameter estimates: [ 1.5994  59.7245 ]
Iteration: 6  Q-function value: 120564.88 parameter estimates: [-0.9599  50.1743 ]
Iteration: 7  Q-function value: 103334.58 parameter estimates: [ 1.2867  43.5483 ]
Iteration: 8  Q-function value:  92669.22 parameter estimates: [-1.1749  38.0216 ]
Iteration: 9  Q-function value:  81572.93 parameter estimates: [-1.6010  36.7623 ]
Iteration: 10 Q-function value:  79838.57 parameter estimates: [-1.9276  36.7629 ]
```

Listing 5.3: MATLAB output of the UTEM algorithm for Example 4.2.

Obviously, the estimates do not converge towards the true parameter values. This indicates that the unscented transformation is not sufficiently accurate to approximate the nonlinear transformation involved in this model.

A major disadvantage of the UTEM algorithm is that the exact quantification of the approximation error is not feasible. This means that there is no simple way to estimate in advance wheather the approximation error will be sufficiently small or not, given the

model structure and the initial parameter estimates. As already shown, the approximation errors occur due to the higher order terms of the Taylor series expansion of mean and covariance of the given distribution. A more detailed understanding of these errors and their influence on convergence would be extremely beneficial, but unfortunately, this is still an open research issue.

5.1.2 The Sequential Monte Carlo EM algorithm: simulation results

The aim of this section is to demonstrate performance of the SMCEM algorithm, which is taken as a reference joint estimation algorithm. It is widely considered as the most versatile for large class of nonlinear dynamic systems. The intention of the simulation study below is to more objectively appraise the potentials of the UTEM vis a vis the reference approach.

SMCEM algorithm - Case II

Consider the state-space model (5.1) with the true parameter values set to $\boldsymbol{\theta}_2$. The Algorithm 4.3 is initialized with $M = 100$ particles and the maximum number of the SMCEM algorithm iterations is set to 100. The maximization (M-step) is performed using the same MATLAB algorithm as in the previous examples. The initial parameter estimates have been chosen to lie somewhere in the region of the explored log-likelihood function (c.f. Figure 5.5).

$$\boldsymbol{\theta}_{SMC1} = [\ a,\ b,\ c,\ d,\ q,\ r\ ] = [\ 0.5,\ 10,\ 8.0,\ 0.05,\ 1,\ 0.1\ ] \tag{5.7}$$

Results of the first 10 iterations are shown in Listing 5.4, while Figure 5.8 displays the trajectory of the current estimates with respect to the parameter values b and q.

```
Iteration: 1
parameter estimates:  [ 0.500 10.000 8.000 0.050 1.000 0.1 ]
Iteration: 2
Q-function value: 280.84 parameter estimates: [ 0.610 14.020 9.433 0.086 1.433 4.2924 ]
Iteration: 3
Q-function value: 284.37 parameter estimates: [ 0.552 14.693 8.254 0.065 1.434 3.4734 ]
Iteration: 4
Q-function value: 259.73 parameter estimates: [ 0.534 15.915 7.774 0.063 1.272 2.2290 ]
Iteration: 5
Q-function value: 203.49 parameter estimates: [ 0.515 17.294 7.608 0.065 1.082 1.5775 ]
Iteration: 6
Q-function value: 152.94 parameter estimates: [ 0.499 18.171 7.540 0.066 0.903 1.1353 ]
Iteration: 7
Q-function value: 102.08 parameter estimates: [ 0.487 18.893 7.486 0.066 0.760 0.8336 ]
Iteration: 8
Q-function value:  54.23 parameter estimates: [ 0.483 19.394 7.434 0.066 0.617 0.6244 ]
```

```
Iteration: 9
Q-function value: -35.41 parameter estimates: [ 0.484 19.680 7.357 0.066 0.513 0.5016 ]
Iteration: 10
Q-function value: -66.72 parameter estimates: [ 0.482 19.922 7.311 0.066 0.440 0.4273 ]
```

Listing 5.4: Output of the SMCEM algorithm - Example 1.

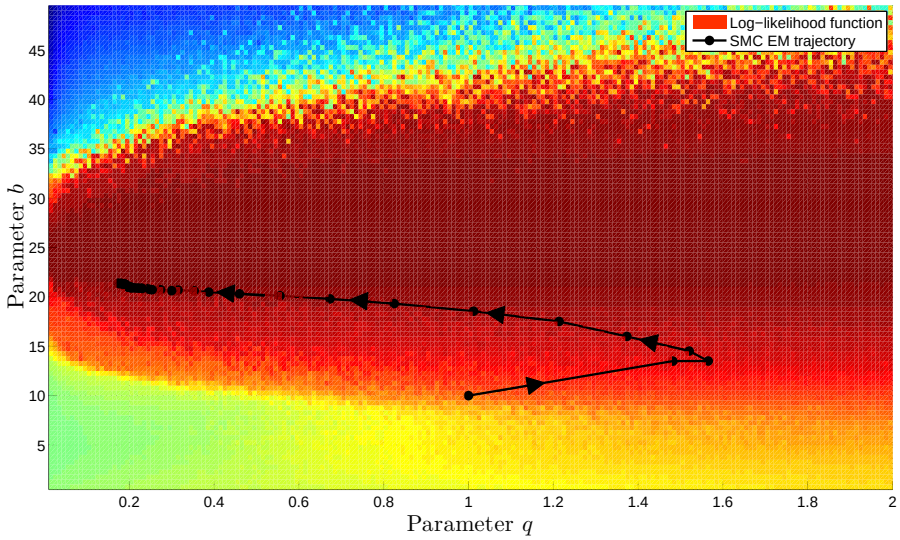


Figure 5.8: Trajectory of the parameter estimates from the SMCEM algorithm - Example 1 projected on the data log-likelihood function in the $b - q$ parameter subspacee.

The parameter estimates after 100 iterations of the SMCEM algorithm for this example were:

$$\widehat{\boldsymbol{\theta}}_{SMC2} = [\ \widehat{a},\ \widehat{b},\ \widehat{c},\ \widehat{d},\ \widehat{q},\ \widehat{r}\ ] = [\ 0.493,\ 21.380,\ 7.317,\ 0.062,\ 0.173,\ 0.320\ ] \tag{5.8}$$

The result indicates that the estimation is numerically stable and seems to converge to the true parameter value. For a better estimate, a larger number of algorithm iterations may be needed.

The Monte Carlo analysis of the SMCEM algorithm

To fully demonstrate the performance of the SMCEM algorithm, a Monte Carlo analysis was conducted. The algorithm was initialized to use 100 particles and a maximum number of EM iterations was set to 1000. A set of 100 different output data realizations $\mathbf{Y}_T$ of length $T = 100$ were generated using the model (5.1) and the initial parameter guess for each estimation was selected randomly from the uniform distribution on the interval corresponding to $\pm 50\%$ around the true parameter value. The individual parameter

estimates for each instance as a function of iteration number are given in Figure 5.9 and the averaged final estimates are summarized in Table 5.3.

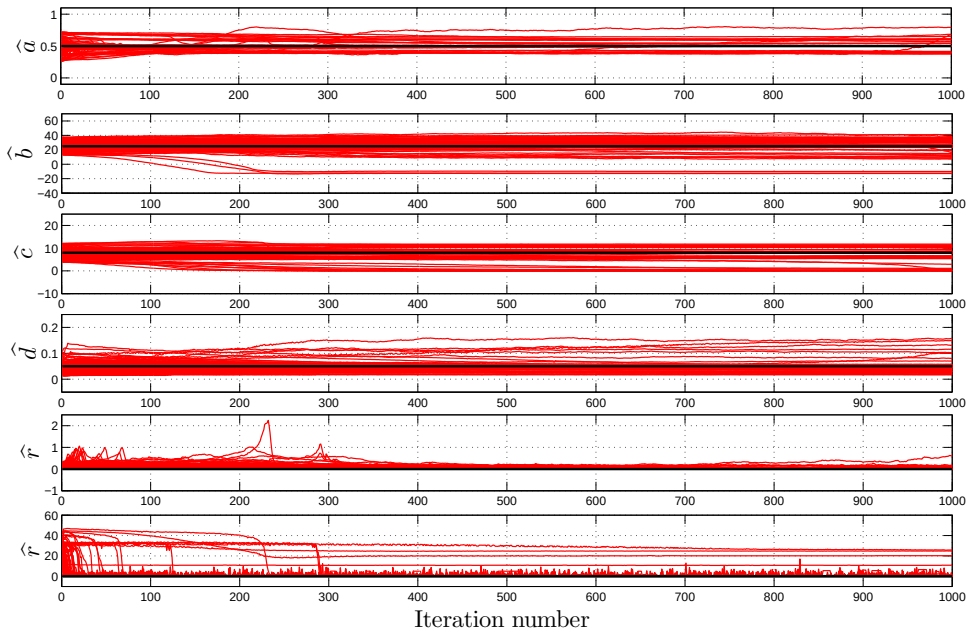


Figure 5.9: The parameter estimates as a function of the iteration number (black line represents the true parameter value).

Table 5.3: The statistics (mean and standard deviation) of the estimates obtained on a set of $N = 100$ Monte Carlo realizations of $\mathbf{Y}_T$.

Parameter	a	b	c	d	q	r
True	0.5	25	8	0.05	0	0.01
Estimated mean	0.507	26.22	7.481	0.049	0.1071	0.1277
Estimate variance	0.081	6.5692	2.1609	0.0195	0.11	0.004

It can also be seen that in some cases, the estimates did not converge to the true value at all which additionally decreases the results statistics.

5.1.3 Comment on the results

This section tried to study the performance of the UTEM and the SMCEM algorithms. UTEM algorithm proved to be computationally very efficient and provides good parameter estimates if the UT approximation converges. In some regions the magnitude of the approximation errors resulted in unstable behavior (the UTEM algorithm - Example 2).

Despite many attractive properties of this approximation method, this may prevent its application in some cases. Additional research in the direction of providing insight into quantification of the approximation error and the convergence criteria is a challenging question.

The second algorithm is based on particle approximation and seems to be more successful in terms of convergence. The SMCEM - Example 1 showed that it can handle more demanding estimation problem, where UTEM algorithm failed. The convergence conditions for the SMC based algorithms were recently established by Hu *et al.* (2008). However, the conditions depend on the number of particles used in the algorithm and the increased number of iterations. Both of them result in a substantial increase of the computational load.

The numerical example demonstrated that system identification of general nonlinear systems is indeed a challenging problem that cannot be solved in a straightforward manner. We believe that both algorithms considered in this section, have appealing properties and that the appropriate version of the algorithm has to be selected individually for each problem.

5.2 Simulated higher-dimensional system

For the sake of better insight into the performance of the UTEM algorithm a four-dimensional state space model of wastewater treatment plant will be used. The model contains four unknown parameters in the state transition part and four parameters of the covariance matrix. Simulated realizations of the states and output trajectories will be used.

5.2.1 Brief description of the process

The process consists of two anoxic reactors, two aerobic reactors and an additional reactor, where water is collected before returning as an internal recycle or passing down to the settler (Figure 5.10). In the pilot plant the biomass (microorganisms) is attached to the small free-floating plastic carriers that are added in the reactors (Ødegaard *et al.*, 1994). Mixers are installed into the anoxic reactors to maintain mixing, while the aerobic reactors are mixed by the airflow. The influent to the pilot plant is waste-water after mechanical treatment, which is pumped to the pilot plant.

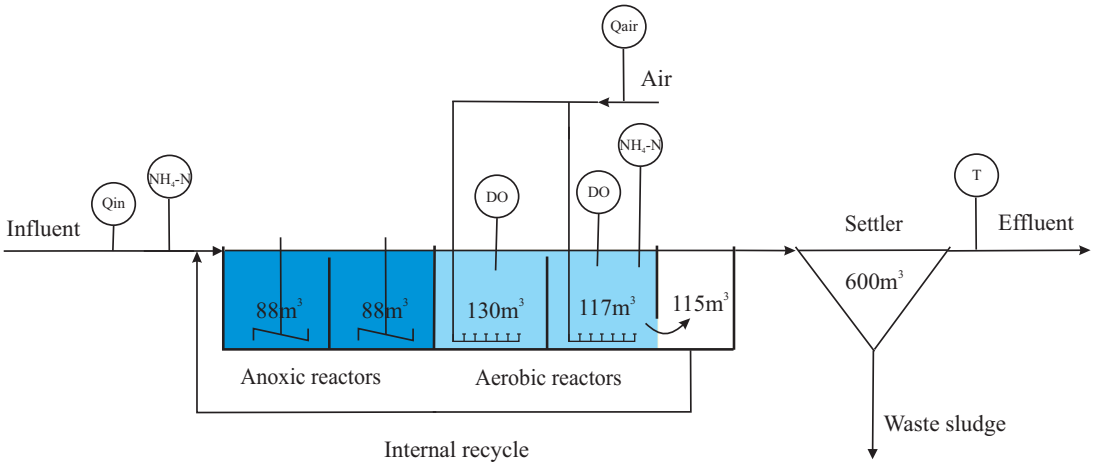


Figure 5.10: Scheme of the waste-water treatment plant.

5.2.2 Mathematical model

A reduced nonlinear model of the process takes the following form:

$$\dot{\mathbf{x}} = -\mathbf{D}\mathbf{x} + \mathbf{F} + \boldsymbol{\rho} + \mathbf{w} \tag{5.9a}$$

$$y = \mathbf{H}\mathbf{x} + \mathbf{v} \tag{5.9b}$$

where $\mathbf{x}$ is the state vector and y is the model output, $\mathbf{w}$ and $\mathbf{v}$ are mutually independent stochastic processes with covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ respectively, $\mathbf{D}$ is the dilution rate matrix, $\mathbf{F}$ is the feed rate vector, $\boldsymbol{\rho}$ is the ammonia nitrogen reaction rate vector and $\mathbf{H}$ is the output vector. Vectors and matrix are defined as follows:

$$\mathbf{x} = \begin{bmatrix} S_{NH1} & S_{NH2} & S_{NH3} & S_{NH4} \end{bmatrix}^T \tag{5.10a}$$

$$\mathbf{F} = \begin{bmatrix} D_{in}S_{NHin} & 0 & 0 & 0 \end{bmatrix}^T \tag{5.10b}$$

$$\boldsymbol{\rho} = \begin{bmatrix} 0 & \rho_2 & \rho_3 & 0 \end{bmatrix}^T \tag{5.10c}$$

$$\mathbf{D} = \begin{bmatrix} D_1 & 0 & 0 & -D \\ -D_2 & D_2 & 0 & 0 \\ 0 & -D_3 & D_3 & 0 \\ 0 & 0 & -D_4 & D_4 \end{bmatrix} \tag{5.10d}$$

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T \tag{5.10e}$$

$$D_{in} = \frac{Q_{in}}{V_1}, D_1 = \frac{Q_{in} + Q_r}{V_1}, D_1 = \frac{Q_{in} + Q_r}{V_1} \tag{5.11a}$$

$$D_2 = \frac{Q_{in} + Q_r}{V_2}, D_3 = \frac{Q_{in} + Q_r}{V_3}, D_r = \frac{Q_r}{V_1} \tag{5.11b}$$

$$\rho_i = r_{NH} \left(\frac{S_{NHi}}{K_{NH} + S_{NHi}} \right) \left(\frac{1}{1 + e^{-k_1 S_{O_i} + k_2}} \right) \Theta^{T-20^oC} \tag{5.11c}$$

The process and measurement covariances values are composed as

$$\mathbf{Q} = \begin{bmatrix} q1 & 0 & 0 & 0 \\ 0 & q2 & 0 & 0 \\ 0 & 0 & q3 & 0 \\ 0 & 0 & 0 & q4 \end{bmatrix}, \qquad \mathbf{R} = [r]$$

The description of the model parameters, known inputs and outputs is given in Table 5.4 (Stare *et al.*, 2006).

Table 5.4: Description of model inputs, outputs and known parameters.

	Symbol	Description
Inputs	S_{NHin}	Ammonia concentration in influent
	S_{O2}	Oxygen concentration in the second reactor
	S_{O3}	Oxygen concentration in the third reactor
	T	Waste-water temperature at the effluent
Outputs	S_{NH3}	Ammonia concentration in the third reactor
Known parameters	V_i	Volume of the i-th reactor
	Q_{in}	Influent flow rate
	Q_r	Recycle flow rate
	Θ	Waste-water temperature coefficient

Other model parameters are unknown and have to be estimated from the input-output data. They are summarized in Table 5.5.

A discrete-time model of the process is obtained by discretization of (5.9) using the first order Euler forward method. Sampling time was set to $T_S = 5$ minutes. The discretized reduced nonlinear model can be written in the following form:

$$\mathbf{x}_{k+1} = (\mathbf{I} - T_S\mathbf{D})\mathbf{x}_t + T_S(\mathbf{F} + \boldsymbol{\rho}) + \mathbf{w}_t \tag{5.12a}$$

$$y_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_t \tag{5.12b}$$

where $\mathbf{x}_k$ is the state vector; y_k is the model output; both in the k-th time instance; vectors $\mathbf{F}$, $\boldsymbol{\rho}$, $\mathbf{H}$ and matrix $\mathbf{D}$ are the same as defined in equations (6.5) and (5.11); $\mathbf{w}_t$ and $\mathbf{v}_t$ are mutually independent stochastic discrete-time processes with covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, respectively.

Table 5.5: Description of the unknown model parameters.

Symbol	Description
r_{NH}	Reaction rate parameter for ammonia nitrogen removal
K_{NH}	Ammonia half-saturation coefficient
k_1	Parameter of exponential oxygen switching function
k_2	Parameter of exponential oxygen switching function
q_i	i-th diagonal element of process noise covariance matrix
r	Measurement noise covariance

5.2.3 Estimation of the model parameters

The time series used for estimation were obtained by simulation and consist of $T = 100$ samples. The vector of true model parameters is selected as follows:

Table 5.6: True model parameter values.

Parameter	True value
r_{NH}	350
K_{NH}	2
k_1	0.35
k_2	5
q_1	0.1
q_2	0.1
q_3	1
q_4	0.1
r	0.1

The performance of the UTEM algorithm will be demonstrated along several steps. In the first step, part of the parameter estimates are fixed to the true value and the rest are considered unknown (see Table 5.7).

Table 5.7: Unknown model parameters in estimation scenarios.

	Unknown parameters
Case I	K_{NH}
Case II	q_3
Case III	$K_{NH},\ q_3$

In the second step, we conduct a Monte Carlo analysis of the influence of different noise covariance values on estimation of model parameters. At the end, a Monte Carlo

analysis of full parameter estimation problem is performed, while keeping estimates of the covariance parameters fixed.

Case I: Estimation of a single model parameter

The UTEM algorithm was initialized with the scaling parameter $\alpha = 0.7$ and maximum number of iterations was set to 20. The estimation procedure was repeated for 50 different realizations of model outputs. Because the maximum likelihood estimator is a function of these random variables, different output realizations will result in different realizations of the estimator.

The initial guesses of the parameter K_{NH} were selected randomly from the interval [0.5, 3.5]. The individual optimization runs are shown in Figure 5.11 and the final estimate statistics in Table 5.8.

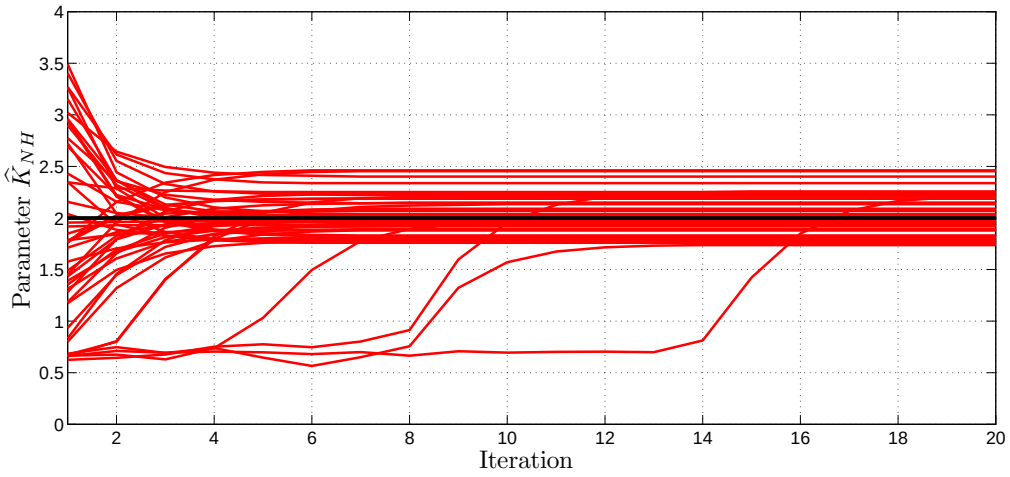


Figure 5.11: Parameter K_{NH} estimates as a function of iteration number (black line represents true parameter value).

Table 5.8: Case I mean values and variance of the final parameter estimates.

Parameter	K_{NH}
True value	2
ML estimate mean	2.08
ML estimate variance	0.23
Converged	100%

It can be seen, that the algorithm has no difficulties locating the true parameter value and the estimates converged independently of the initial parameter guess.

Case II: Estimation of noise parameter

The unknown parameter in this case is the noise variance on the third system state q_3. The UTEM algorithm was again initialized with the scaling parameter $\alpha = 0.7$ and maximum number of iterations was set to 20.

Again, the estimation procedure was repeated for 50 different output realizations. The initial guess of parameter q_3 was selected randomly from the interval $[0.5,\ 1.5]$. The individual optimization runs are shown in Figure 5.12 and the final estimate statistics in Table 5.9.

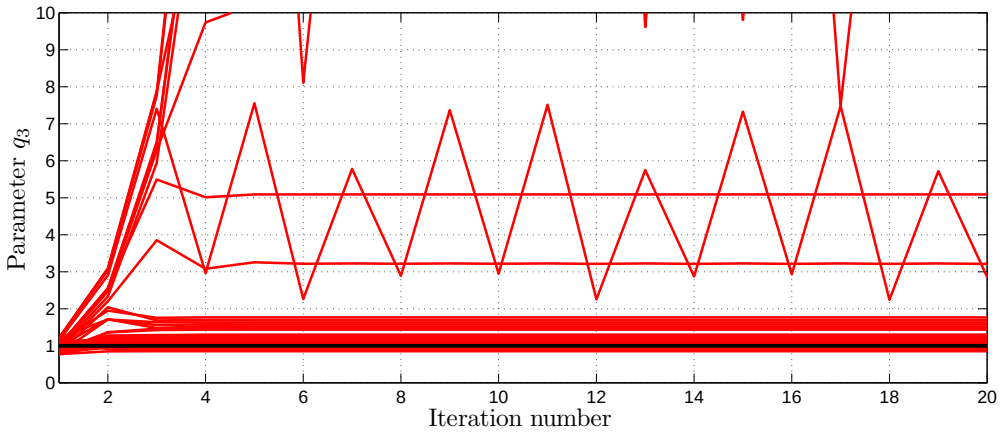


Figure 5.12: Parameter q_3 estimates as a function of iteration number (black line represents true parameter value).

Table 5.9: Case II mean values and variance of the final parameter estimates.

Parameter	q_3
True value	1
ML estimate mean	1.19
ML estimate variance	0.25
Converged	84%

It can be seen, that the estimation of noise parameters poses a more difficult estimation problem for UTEM algorithm. In 16% of the cases, the estimated value differs from the true for more than 100% of the true value. For these cases, we concluded that the algorithm failed to converge to the true value. However, in the cases when the UTEM algorithm converged, the final estimation statistics show a relatively good quality of estimation.

Case III: Estimation of parameter and noise

For the next case model parameter K_{NH} and noise parameter q_3 are estimated. The UTEM algorithm was again initialized with the scaling parameter $\alpha = 0.7$. Maximum number of algorithm iteration was set to 30.

The estimation procedure was repeated for 50 different output realizations with the initial guess of parameter q_3 selected randomly from the interval $[0.5,\ 1.5]$ and parameter K_{NH} from the interval $[1.5,\ 2]$. The individual optimization runs are shown in Figure 5.13 and the final estimate statistics in Table 5.10.

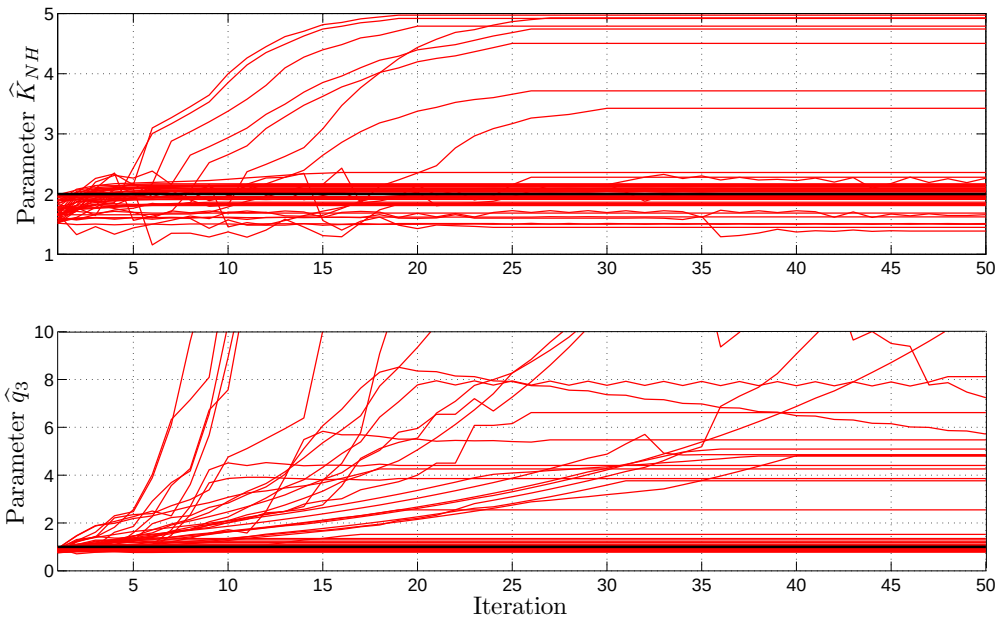


Figure 5.13: Parameter K_{NH} (top) and q_3 (bottom) estimates as a function of iteration number (black line represents true parameter value).

Table 5.10: Case III mean values and variance of the final parameter estimates.

Parameter	q_3	K_{NH}
True value	1	2
ML estimate mean	1.02	2.02
ML estimate variance	0.19	0.29
Converged	48%	80%

It can be seen, that in the case when the UTEM algorithm converged, the final parameter estimates are relatively close to the true values. However, the estimation converged in

only 48% of the cases for parameter q_3 and 80% of the cases for parameter K_{NH}. The divergence of the UTEM algorithm seems to be caused by the noise covariance parameters. Because the log-likelihood is approximated by unscented transformation, the optimum of the approximated log-likelihood with respect to covariance parameters is shifted away from the true location. The results suggest that the estimation of noise parameters is a more demanding problem and leads to very poor convergence of the UTEM algorithm.

The influence of noise covariance on parameter estimation

To further examine the influence of covariance parameters on the estimation, we preform a Monte Carlo analysis of the influence of covariance parameter values on the estimation of other model parameters.

The analysis is preformed in the following way. Each estimation run starts from the same initial parameter guess, but with random values of the noise parameters from the interval corresponding to ±50% of their true values. The maximum number of the UTEM algorithm iterations was set to 50 and the estimation procedure was repeated for 50 different noise parameter values. The distributions of the final parameter estimates for the model parameters are given in Figure 5.14. It is important to note, that the UTEM

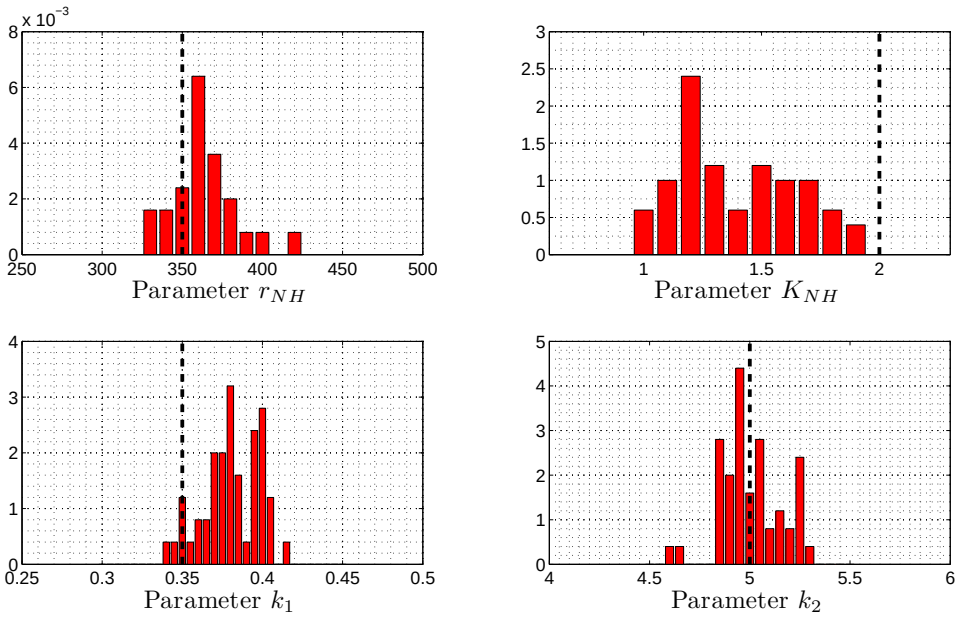


Figure 5.14: Distribution of the final parameter estimates as the result of random combinations of covariance parameters.

algorithm converged in every run. The final parameter estimates are clustered around the true values, but the different noise parameters affect the quality of the estimate.

The Monte Carlo analysis of model parameter estimation

The Monte Carlo analysis of the UTEM performance is limited only to the unknown model parameters. The estimation procedure consisted of 30 iterations and was repeated 100 times with different initial parameter values selected from the uniform distributions on the intervals:

$$\begin{aligned} r_{NH0} &\in [\ 200,\ 400\ ] \\ K_{NH0} &\in [\ 1,\ 3\ ] \\ k_{10} &\in [\ 0.3,\ 5\ ] \\ k_{20} &\in [\ 4,\ 6\ ] \end{aligned}$$

Figure 5.15 shows the parameter estimates in all estimation procedures as a function of iteration number. Table 5.11 shows summary of the results in terms of mean parameter estimate value.

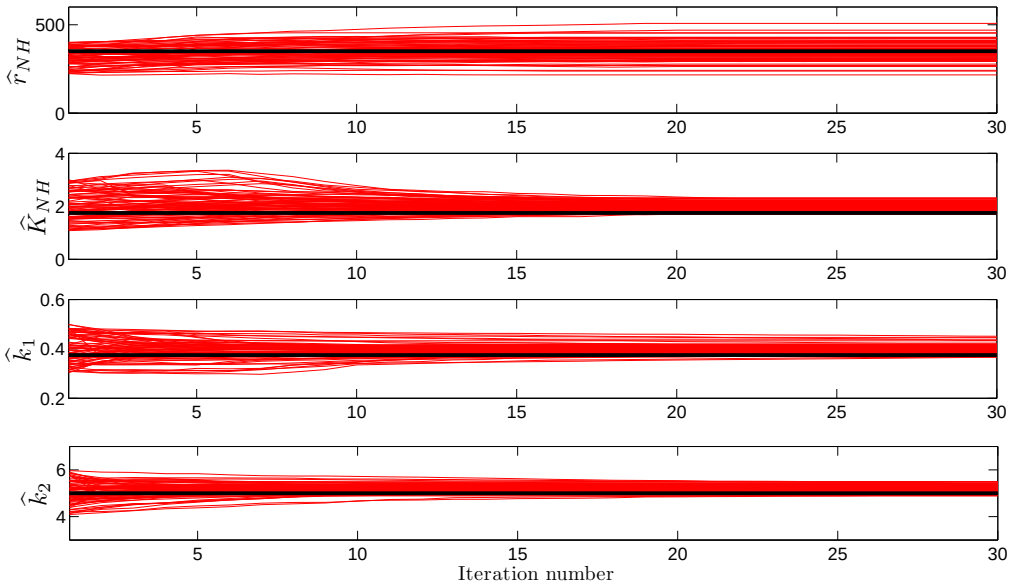


Figure 5.15: Model parameter estimates as a function of iteration number. Black line represents true parameter value.

The estimated model was validated using simulated data. The true and the estimated model were both used to simulate the nitrogen concentration in the fourth anaerobic reactor for the period of 18 days. The model outputs, are shown in Figure 5.16.

It can be seen that the estimated model output is the same as the true model output. The errors are partially consequence of the approximation used in the estimation

Table 5.11: Mean values and variance of the final parameter estimates in Monte Carlo analysis.

Parameter	r_{NH}	K_{NH}	k_1	k_2
True value	350	2	0.35	5
ML estimate mean	340.7	2.03	0.41	5.22
ML estimate variance	70.2	0.203	0.022	0.204

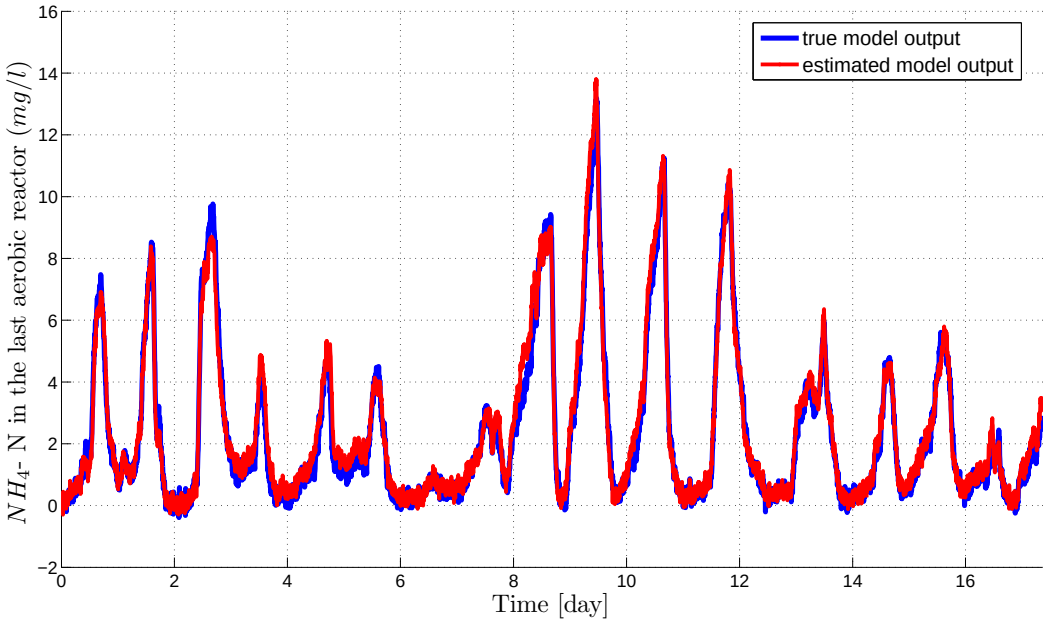


Figure 5.16: Model validation on simulated data.

procedure.

5.2.4 Comment on the results

In this analysis we have shown, that the UTEM algorithm can cope with high dimensional problems in terms of computational load. The challenge in this estimation problem were the covariance parameters. Their estimation greatly decreased the stability of the estimation procedure and resulted in divergence. The reason for this may be, that the noise parameters are more sensitive to approximation errors introduced by the UT. However, the full explanation of this behavior remains a topic for future investigations.

6 Model-based failure prognostics of gear drives

6.1 Background and motivation

Operational safety, cost effective maintenance and asset availability have direct impact on the competitiveness of industrial organizations. Surveys show, that maintenance costs can represent 4% - 8% of the companies income. These figures can vary significantly, depending on the sector. On top of that, one third of these expenses are wasted on unnecessary or inappropriate maintenance (Heng *et al.*, 2009). The introduction of new complex and advanced machines threatens to increase these numbers even more. Therefore, there is a clearly visible need for continuous development of more effective maintenance strategies.

The simplest form of maintenance is reactive maintenance, where no action is taken until the component fails and is then repaired or changed with a new one. This type of maintenance is only appropriate for smaller and cheaper items of equipment which do not affect entire production in the case of failure. To monitor crucial assets, preventive maintenance was introduced. It involves periodic inspections of components, regardless of their health. This strategy increases the reliability of operation, but it is clearly not efficient in terms of costs. It was reported, that 99% of failures are preceded by some sort of noticeable indicators (Block and Geitner, 1997). Therefore, to optimize costs, condition-based maintenance is the most prominent option. With this strategy, the component is constantly monitored and when fault is detected and localized already in its incipient phase.

An important emerging feature of new generation of condition monitoring systems enables prediction of future evolution of the fault and thus enables the plant personnel to accommodate maintenance actions well in advance. Even more, it can predict the remaining useful life of the component under changing operating condition, thus providing information to operators on how the different operating regimes will lengthen or shorten the components useful life. This is a relatively new research area and has yet to receive its prominence compared to other condition monitoring strategies (Heng *et al.*, 2009).

6.1.1 Objectives

Our focus will be on mechanical drives. They are the most ubiquitous item of equipment in manufacturing and process industries as well as transportation. During the operational life-cycle, these items are subjected to wear, fatigue, cracks and other destructive processes. These processes can not be directly observed or measured without interrupting the operation of the machine. The extent of the damage has to be inferred from the available signals, which are usually vibrations, acoustic emissions, oil contaminants, etc. The objective our work is to develop the methods that will be able to predict the failure of the mechanical drive and estimate its remaining useful life based on easily accessible signals.

6.2 Gear failure modes

During operation gears are subjected to complex condition deterioration processes related to stress and wear. The notion "failure" should be understood in the underlying context as degraded condition rather than state with lost functionality. The three most frequent types of gear failures are (Deters, 2009)

- surface fatigue failures,
- failures due to wear and
- failures due to scuffing.

Surface fatigue

Load is transmitted from one gear to another over contacts formed by teeth. Repeated application and removal of load on tooth surface results in fatigue. The major part of these faults are initiated below the surface at various depths. However, as the fatigue progresses, the surface of the teeth gets damaged.

The life span of the gear due to fatigue depends on load and the operating time (working cycles). There are two stages of fatigue failure: initial pitting and spalling.

Initial pitting appears over time due to localized concentrations of overload, which can cause small pits distributed either uniformly over the surface at the pitch line or just locally at one end of the tooth. It only progresses if the localized overload conditions are present. During evolution of the initial pitting the edges of the pit deform plastically and tend to smooth out, thus leaving the impression of a self-corrective process. Initial pitting is usually not conceived as failure.

If the tooth surface is subjected to high surface stress and high sliding velocities, small cracks appear in the toots surface. This is the process of spalling. These cracks gradually spreads until a piece of the material is removed from the surface.

Wear

The basic mechanism causing wear is insufficient lubricant film thickness allowing surface-to-surface contact between the matching surfaces of the teeth. Other factors can contribute to wear such as abrasive particles in the lubricant or corrosion of the tooth surface.

Scuffing

Normally, scuffing occurs when a combination of load conditions, oil temperature and sliding velocities cause breakdown of the oil film. The oil film is present between the two teeth surfaces. Hence a metal to metal contact occurs, due to which the welding of the surface asperities can occur, subsequently followed by breaking of the weld.

6.2.1 An overview of the existing failure prognostic techniques

Development of prognostic models that can be used to predict the remaining useful life of the items of equipment has attracted a significant amount of research in recent years. According to the (ISO 10825, 1995), the term prognostics corresponds to estimation of the time to failure and risk for one or more existing and future failure modes. Several good overview papers addressing the prognostic techniques exist the most recent being those of Heng *et al.* (2009); Sikorska *et al.* (2010).

Existing methods for predicting failures in rotating machinery can be grouped around three major concepts:

- approaches based on event data records,
- approaches relying on condition data and
- combination of both approaches.

The approaches belonging to the first group rely on accumulated data records from large number of same items of equipment or on large set of repeated experiments in specified laboratory conditions. Time-to-fail estimates and other failure probability rates can be described by statistical failure models like Weibull or log-normal distribution (Elsayed, 1996; Vachtsevanos *et al.*, 2006).

The second group is a rich set of approaches that can roughly be divided into physics-based models and data-based models. The physics-based models rely on detailed physical

modeling by means of finite element method, which serves to compute spatial distributions of stresses in the material (Chaari *et al.*, 2009; Fajdiga *et al.*, 2003). The progression of local cracks in the material is computed by locally applying the Paris' law (Suresh, 1998).

The idea of data-driven methods is to make use of condition monitoring data to build the model and then use the model to predict future trend. Many ideas were proposed, perhaps the most simple one is to describe the condition monitoring data as static functions of time. Model parameters are adapted on-line so that change in the trends can be promptly traced (Cempel, 1987).

A number of authors suggested use of neural networks as an efficient time series prediction tool. For example, Wang *et al.* (1999) apply neural network to predict crack in rolling bearing by on-line adaptation of the network parameters. Unfortunately, only one-step ahead prediction is achieved. Problems might also occur when the training data set is too short. To partly alleviate this problem Wang *et al.* (2004) combine neuro-fuzzy predictor with expert knowledge needed to tune the predictor. Some authors have also suggest the use of nonlinear state filtering and prediction based on various particle filters strategies . Their idea is to calculate the probability density function of the predicted life times, based on a dynamic model with known parameter values (DeCastro *et al.*, 2009; Orchard and Vachtsevanos, 2009).

The combined approaches tend to make efficient use of reliability data in the prognostic models. Part of the algorithm set is devoted to acquisition of the vibro-acoustical features, which are used to identify specific points at which the transition from one failure mode to another occurs. Calculations take into account knowledge of the mechanical component, historical failure mode, load profiles and other data related to operation. Final assessment of the remaining useful life results from comprehensive treatment of the current condition and historical records.

6.3 Vibrational model

The idea presented in this dissertation falls within the class of data driven grey-box models. The idea is to adopt the model structure based on physical modeling and then on-line update the model parameters based on available condition monitoring data. The novelty lies in using on-line parameter and state estimation as the key operations in model updating. Updated stochastic model is used to calculate the pdf's of times at which the system reaches critical condition.

A model of gear pair is shown in Figure 6.1. The conceptual setup consists of a driving gear with rotational speed ω_1 and number of teeth N_1 and the driven gear with the number of teeth N_2. The rotational speed of the driven gear is $\omega_2 = \frac{N_1}{N_2}\omega_1$ and meshing frequency

(rate at which gear teeth mate together) reads $\omega_g = \omega_1 N_1$

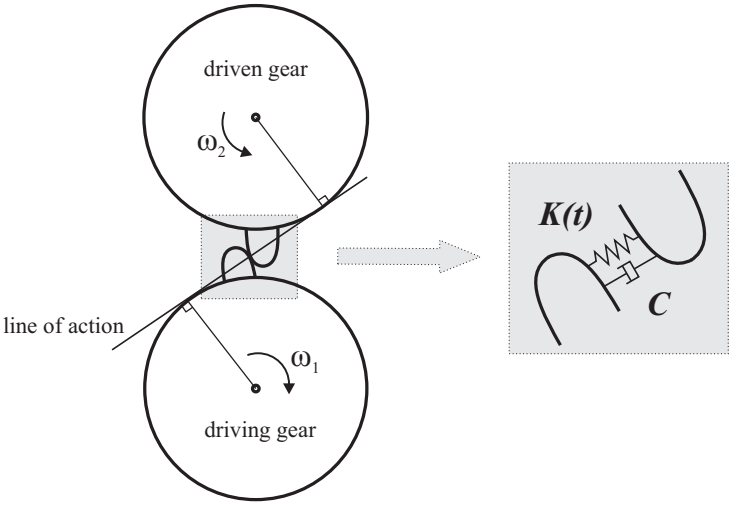


Figure 6.1: The dynamic model of a pair of gears

Several authors have studied the relationship between surface pitting and wear condition on system vibrations (Choy *et al.*, 1996; Wang and Zhang, 1998; Yibo *et al.*, 2010). The pair of spur gears is considered as a stochastic system with one degree of freedom. Forces on the teeth act along the line of action as depicted in Figure 6.1. The force F is assumed to consist of the three components: a static force F_0 between the gears along the line of action and F_e staying for manifestation of faults and errors in gears.

The equation of motion of the gear pair is given by

$$m\ddot{x} + C\dot{x} + K(t)x = F_0 + F_e \tag{6.1}$$

where m denotes equivalent mass of the a gear system and is calculated as $m = m_1 m_2/(m_1 + m_1)$ with m_1 and m_2 being masses of the gears, C is damping coefficient mostly depending on the lubricant, $K(t)$ is time varying gear-mesh stiffness. The force F_e can be viewed as

$$F_e = K(t)e_p(t) + K(t)e_r(t) \tag{6.2}$$

where e_p is relative displacement, which is a periodic signal caused by misalignment and imbalance errors on one side and condition of the teeth surfaces (meshing errors) on the other side. The component e_r represents relative displacement caused by random effects. The component F_0 represents the average of the displacement distortion ($F_0 = \overline{K(t)e(t)}$).

The periodic displacement can be represented as a function of the gear-mesh ω_g and rotational speed ω_1

$$e_p(t) = a \sin \omega_1 + \sum_{n=1}^{\infty} b_n \cos\left(n\omega_g t + \phi_n\right) \tag{6.3}$$

where a is the amplitude of the spectral component at ω_1, b_n is the amplitude of the n-th harmonic of gear mesh frequency and ϕ_n is its phase.

Force caused by meshing gears excite the entire gearbox transmission structure up to the vibrational sensor. The measured vibration signal $z(t)$ can be viewed as convolution between the structure transfer function and force $M\ddot{x}(t)$. What is important to stress is that the spectral content of vibrational signal $z(t)$ is characterized by the spectral content of the excitation side in (6.1).

It should be noted that change in the coefficients b_i implies changes in the gear-mesh component and its multiples. Consequently, a change in feature y occurs. For example, if feature is defined as the vibrational spectral component at the gear-mesh frequency, then y directly depends on the coefficient b_1. Since the evolution of the pitting process is impossible to describe accurately, it seems to be a good option to view it as a stochastic process.

The coefficient b_1 in (6.3) can be modeled as result of a stochastic process. The key conjecture made in this work is that the stochastic parameter b_1 depends on the condition of the damaged surface. When pitting and later spalling progress, an increased amount of debris is produced thus increasing the roughness of the surface. Larger is the pitted area, more irregular is the relative displacement e_r, which leads to increased vibrations.

Various laboratory studies have been conducted to find out the progression of pitting and spalling areas under various load, lubricant and temperature conditions with strong focus on bearings. For that component Kotzalas and Harris (2001) proposed a spall progression rate $\frac{dS_{sp}}{dt}$ which is as follows

$$\frac{dS_{sp}}{dt} = c\left((\sigma_{max} - \sigma_{av})\sqrt{\pi S_p}\right) \tag{6.4}$$

where σ_{max} and σ_{av} denote maximal stress and average shearing stress and S_{sp} is the spall size. It should be noted that the model is derived and validated on laboratory conditions using a special test rig. However, the direct applicability of the model in realistic conditions is rather questionable since none of the parameters as well as spall size are not directly available.

In this work the same concept will be adopted for progression of pitting and spalling in gears. Though the mechanisms differ from those in bearings, the progressive nature of the processes are similar. Instead of a specific model structure (6.4) a more general structure with minimal parameterization will be employed

$$\frac{dS_{sp}}{dt} = f(S_{sp}, \theta(t)) \tag{6.5}$$

where a time-varying parameter θ reflects the variable stiffness properties. Based on (6.5), we can define two state variables. First one corresponds to the spall progression S_{sp} and

the second one to the time-varying parameter θ. Two versions of the state-space model are considered in this work:

- **Black-box linear model**. For this model we consider a time-varying parameter as a separate system state in addition to spall size. Additionally, instead of providing an exact description of the function $f(\cdot)$ we assume linear dynamics with unknown parameters. The rationale behind this model is that this type of model serves as a local approximation to the actual nonlinear dynamics.

- **Black-box nonlinear model**. Instead of local linear model, we assume a population-based growth model (Orchard and Vachtsevanos, 2009) with unknown parameters. Instead of approximating local dynamics, this model is a nonlinear approximation of the entire function $f(\cdot)$.

The models both have the same output, which is the feature value y and no measured inputs. The black-box nature of both models removes the need for case dependent derivation of exact relation between feature values and faults and thus increases the applicability of the proposed approach.

6.4 The experimental set-up

For the purpose of development and verification of the model-based prognostics tools, the experimental test bed has been used (Figure 6.16). It consists of a motor-generator pair with a single stage gearbox. The motor is a standard DC motor powered through a Simoreg$^{\text{TM}}$ DC drive. A generator is being used as a break and the generated power is being fed back in the system, thus achieving the breaking force. For the purpose of predicting the remaining useful life (RUL) of gears, all other components (e.g. bearings, shafts) that have been used are of higher quality, therefore minimizing the influence of other faults on the measured signals (Suhač *et al.*, 2008).

The most informative and easily accessible signals that offer information on gear health are vibration signals (Combet and Gelman, 2009). Vibration signals are acquired from eight accelerometers placed on all the main components (gearbox, input shaft, output shaft, motor bearing) and in different orientations (x, y, z-axis). The schematics plot of sensor placement is shown in Figure 6.3.

The placement of the individual sensor determines its sensitivity of vibrations with respect to damaged gear surfaces. For example, sensors positioned closer to the gearbox are more influenced by the underlying faults. Also, sensors on the shafts are more sensitive

Simoreg$^{\text{TM}}$is a registered trademark of Siemens AG.

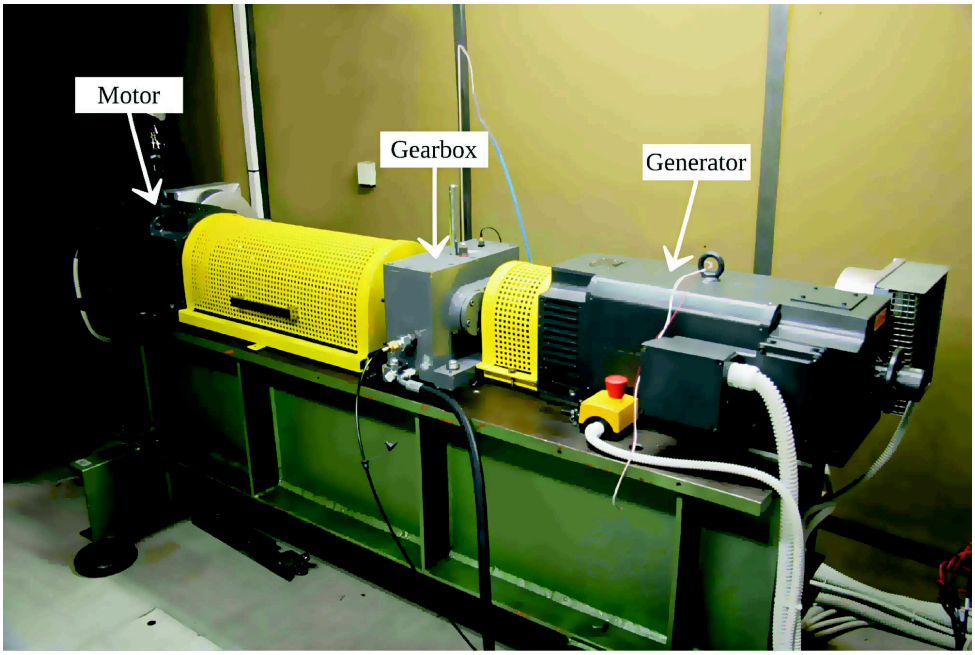


Figure 6.2: The experimental test bed.

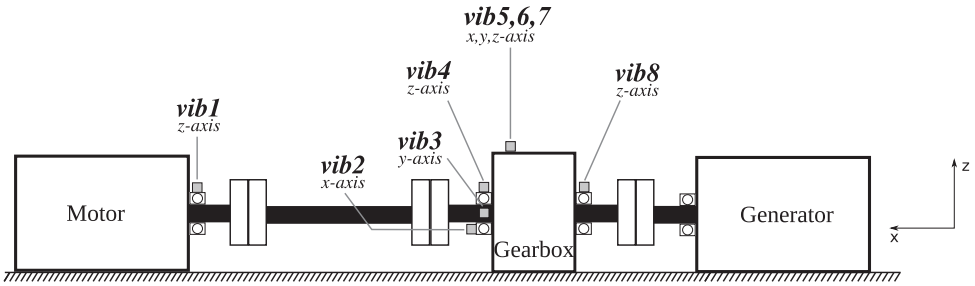


Figure 6.3: Vibration sensors placement scheme (sensors are named *vib1* to *vib8*).

than those placed on the gearbox frame. Therefore, the quality of the sensor data from all sensors has been compared in terms of signal to noise ratio. According to this comparison, only signals from sensors with the highest sensitivity have been used in further analysis. In the presented setup, the sensors labeled *vib4* and *vib8* in Figure 6.3, which measure the vibrations on gearbox output and input shafts respectively, have proved to be the most sensitive to gear damage.

6.4.1 The experimental protocol

The data used in this study was collected from experiments conducted on the test bed. The test run was done with a constant torque of 82.5Nm and constant speed of 990rpm. This speed of 990rpm generates gear mesh frequency (GMF) f_g = 396Hz, rotational

speed of driving shaft $f_1 = 16.5$Hz, and rotational speed of driven shaft $f_2 = 24.75$Hz. The signals were sampled with sampling frequency $f_s = 80$kHz. Each acquisition session took 5 seconds. The acquisition was repeated every 10 minutes as illustrated in Figure 6.4.

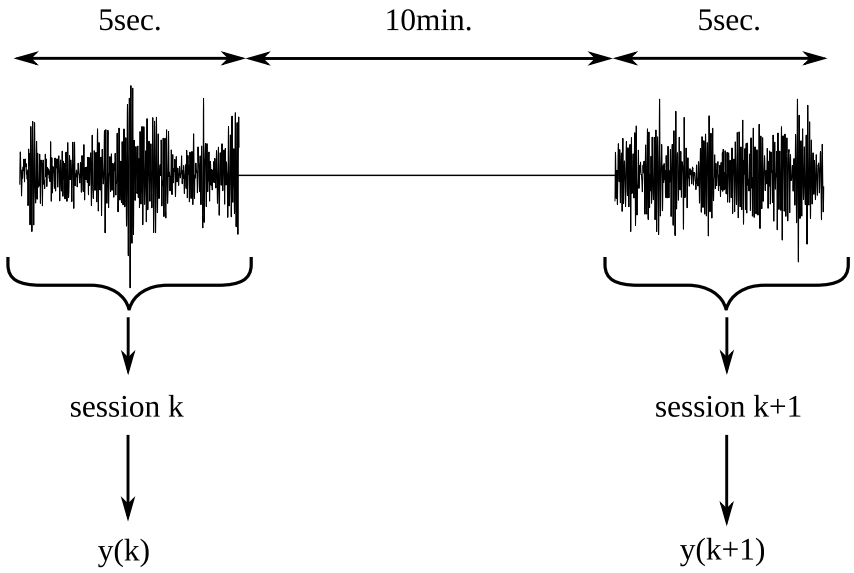
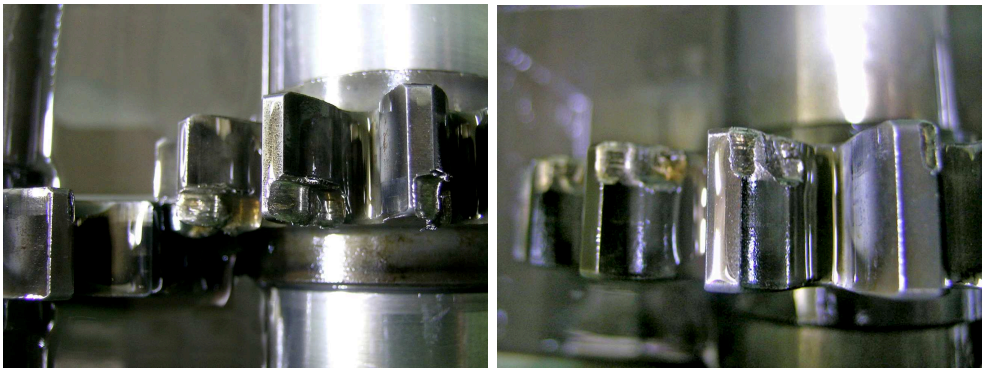


Figure 6.4: The concept of signal acquisition.

Under normal operational conditions, gears have a relatively long useful life, therefore the accelerated damage mechanisms have to be used in order to speed up the experiment. In our case, the contact surface between the gears was decreased to 1/3 of the original surface. In this manner the fault evolution horizon is made shorter. The relative position of the gears can be visible in Figure 6.5.



(a) Output gear (b) Input gear

Figure 6.5: Output and input gear at the end of the experiment with strong plastic deformations of the teeth.

The overall experiment run took 65 hours. At the end, both gears were heavily damaged i.e. on both gears spalling can be seen on all teeth, which progressed eventually into

a plastic deformation of some of them, as shown in Figure 6.5.

The validation run was performed with normal setup (without the accelerated damage mechanism) and the entire duration of the experiment was 420 hours. The evolution of the gear damage in this more realistic case is illustrated in Figures 6.6, 6.7 and 6.8.

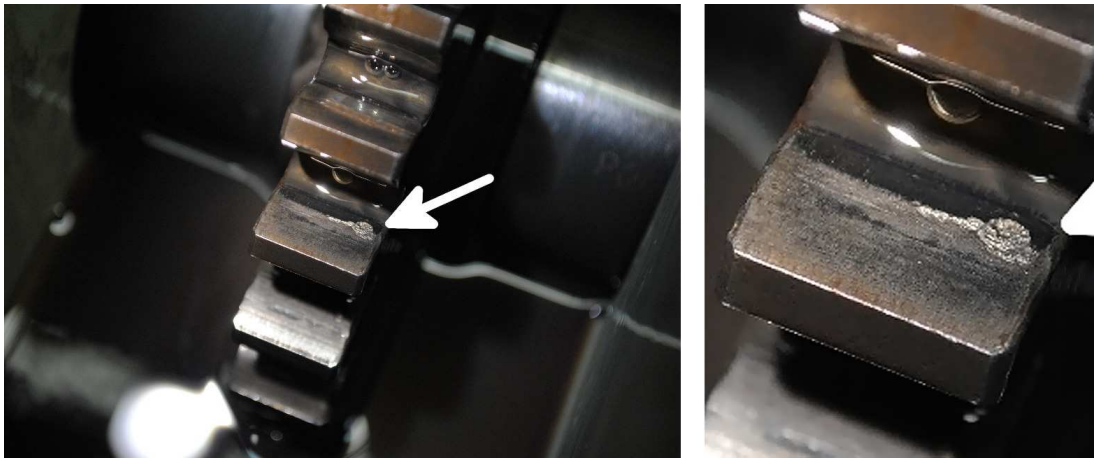


Figure 6.6: Gear condition after 48 hours of operation.

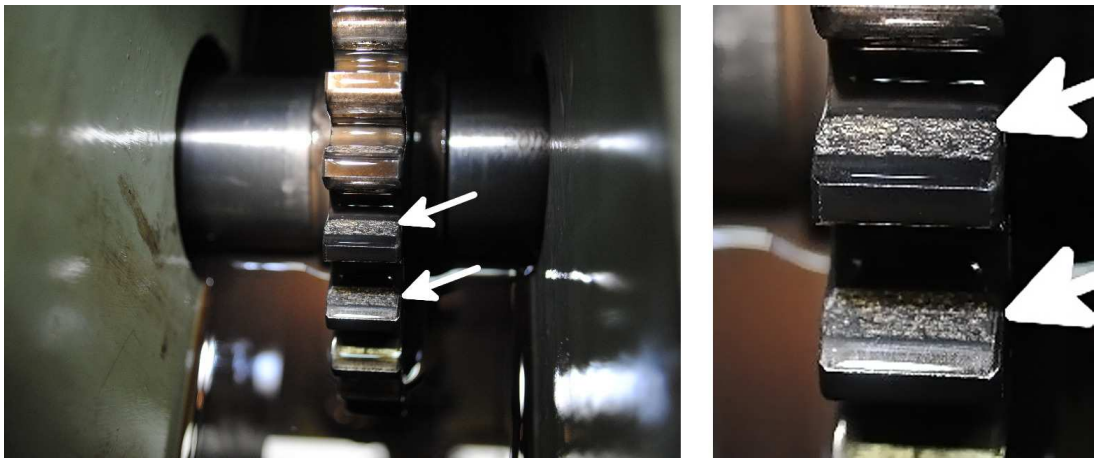
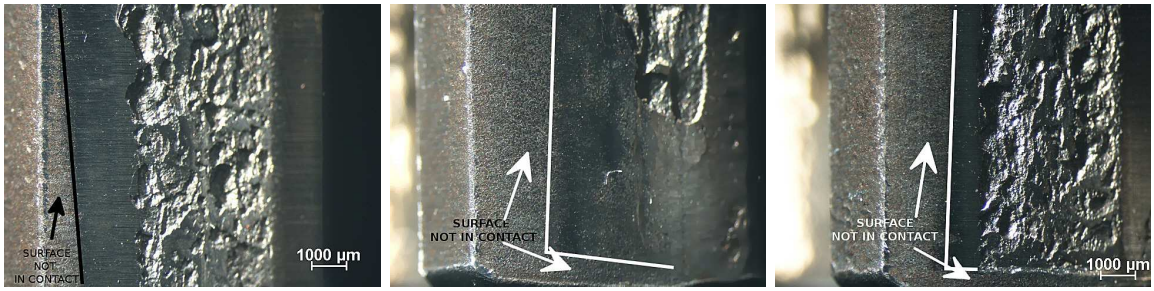


Figure 6.7: Gear condition after 360 hours of operation.

6.4.2 Feature extraction

Signals from sensors were acquired simultaneously during acquisition session and analyzed using envelope spectra (Ho and Randall, 2000). Since the best quality data were obtained from *vib4* and *vib8* only the feature vectors $y_4(k)$ and $y_8(k)$ were used in further analysis. From each sensor, at each acquisition session k, a feature vector $[y_4(k), y_8(k)]$ was derived. Each element of the feature set represents power density at the gear-mesh frequency in



(a) (b) (c)

Figure 6.8: Microscope images of the damaged teeth of the gear.

the envelope spectrum. The complete record of the feature values, at the end of the experiment, consists of two time series, each corresponding to one vibration sensor.

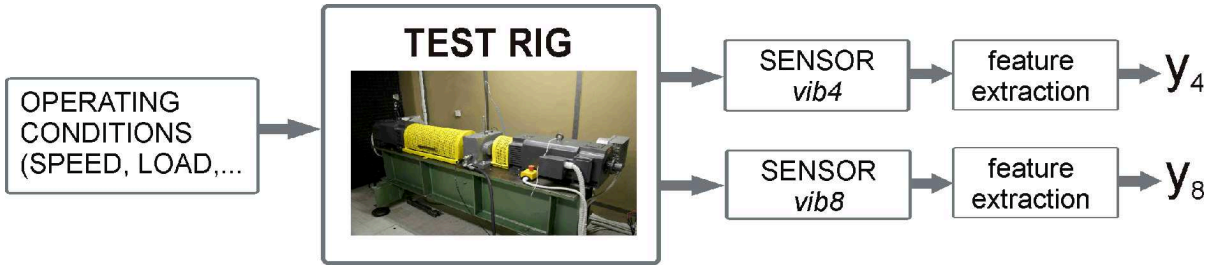


Figure 6.9: Feature extraction procedure.

6.5 Dynamic models of gear condition

Time series derived from vibration signal is modeled as an output of a discrete time, stochastic dynamic system and the current condition of the machine is described with nonlinear state-space model with the following form:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{w}_t, \boldsymbol{\theta}) \tag{6.6a}$$

$$y_t = g(\mathbf{x}_t, v_t, \boldsymbol{\theta}) \tag{6.6b}$$

where y_t is a system output (i.e. feature value, derived from one vibration sensor), $\mathbf{x}_t$ an unmeasured system state vector, $\mathbf{w}_t$ is an i.i.d. random process, v_t measurement noise and $\boldsymbol{\theta}$ a vector of model parameters.

If linearized, (6.6) takes the form

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{w}_t \tag{6.7a}$$

$$y_t = \mathbf{C}\mathbf{x}_t + v_t \tag{6.7b}$$

We assume that system dynamics starts with an initial (healthy) state vector $\mathbf{x}_0$ with mean $\boldsymbol{\mu}_0$ and covariance matrix $\mathbf{P}_0$. The observed system output data, indexed by time $\mathbf{y}_t$, represent the time series we wish to analyze or predict.

An important thing to note when using this type of models is that since fault progression will increase in time, the resulting dynamical model will turn out to be unstable. This in turns means, that the model parameter estimates will not be consistent (Ljung, 1996). The loss of consistency means, that the parameter estimates will not converge to their true values as the number of samples will increase. However, since we are working with the black-box models and are interested only in relatively short time prediction we conclude, that the inconsistent parameter estimates are suitable for the purpose thy are used in this work.'

This is only one possible way to detect and predict failures based on measured vibrations or vibration feature values. For example, a more straightforward way would be to use linear regression to detect current trend. However, the motivation for state-space model lies in its possible extensions, mainly the inclusion of the operating conditions as a measured input. The operating conditions affect the feature values both directly and indirectly via the increased fault progression.

The purpose of both models is the estimation of the current state and prediction of the remaining useful life of the machine. Distribution of the RUL is estimated based on first passage time of the feature value. The first passage time (FPT) of a random process is determined as the time instance when the random variable reaches a specific value for the first time. The rationale for the FPT approach is, that in mechanical systems the feature value is related to the vibration level, whose admissible values are prescribed by standards. Therefore, the feature value must not exceed the critical value.

6.6 Linear model of gear degradation process

The simplest form of the state-space model for modeling the degradation processes in gears is a linear black-box model. Although the actual degradation dynamics can be far from linear, the short-term dynamics are approximately linear. To include the nonlinear nature of the underlying process in to model, we estimate the model parameters based on windowed data.

The vibration feature time series is predicted in the following way. At time t, we estimate the parameters of the underlying stochastic model (6.7) based on the data window $\mathbf{y}_{(t-N_w+1):t}$, where N_w is the window length. As the rate of wear will change over time, the values of the model parameters will change as well. The estimated model parameters will thus determine the current trend in the feature values, while the noise covariance

parameters comprise the influence of the varying noise component (which changes as the damage progresses).

The estimated model, obtained from each time window, is then used to predict the future behavior, as well as the time t^* such that $\mathbf{y}_{t^*} \geq y^*$ for the first time.

This section we will present the algorithm setup, tracking of the model parameters, prediction of the RUL based on analytical results and the Monte Carlo analysis and validation of the results.

6.6.1 Model identification

For the linear black-box model, the dynamic state-space model of the degradation process takes the form of (6.7) where

$$\mathbf{x}_t = \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix}^T \tag{6.8a}$$

$$\mathbf{y}_t = [\ y(t)\ ] \tag{6.8b}$$

$$\mathbf{w}_t = \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{Q} \right) \tag{6.8c}$$

$$v_t \sim \mathcal{N}(0, r) \tag{6.8d}$$

and the vector of unknown model parameters is given as

$$\boldsymbol{\theta} = [\ \mathbf{A},\ \mathbf{C},\ \mathbf{Q},\ r\ ] \tag{6.9}$$

It is important to note that the system states ($x_1(t)$ and $x_2(t)$) in this model do not have any physical meaning. The system states serve only to describe the dynamic behavior of the feature values. However, using the appropriate transformation of the system states, the state-space model can be transformed into different canonical forms (Ljung, 1999), where the system states would correspond to the actual health of the gears.

6.6.2 On line tracking of model parameters

With the presented problem setup, the implementation of the EM algorithm for parameter estimation (Algorithm 4.1) is straightforward. As soon as a new sampling session is over and a new value of the vibration feature is collected, the Algorithm 4.1 is used to compute current model parameters estimates using the last 100 samples. The time window $N_w = 100$ samples corresponds to approximately 18 operating hours. It was chosen ad-hoc based on heuristic prior knowledge. A time window that is to short might cause strong fluctuations of parameter estimates and frequent changes in predicted outputs due

to accentuated impact of random fluctuations in y_t. Too long window might significantly increase detection delay.

In addition to the window length, the initial values of the model parameters also have to be selected prior to the start of the estimation. One of the advantages of using a local linear model is that the estimation result will converge to the true optimum, irrespective of the initial parameter values Gibson and Ninness (2005); Shumway and Stoffer (2005). The initial parameter values only influence the number of steps needed by the algorithm to achieve the convergence criteria. The following configuration of initial values was selected.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 1 \end{bmatrix}, \boldsymbol{\mu}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{P}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (6.10)$$

The convergence of the EM procedure is achieved and the algorithm is stopped when the relative increase of the log likelihood function is less than 10^{-4}.

The algorithm was first tested on the time series corresponding to the measurements of the 8^{th} vibration sensor (*vib8* in Figure 6.3), which measures the vibrations in the z - direction (c.f. Figure 6.3) on the output shaft. Due to the position of the gears, the impacts between the gear teeth cause the strongest vibrations in z - direction. Therefore the feature values from this particular sensor are expected to be the most informative with regard to gear health. The *vib*8 feature time series is shown in Figure 6.10

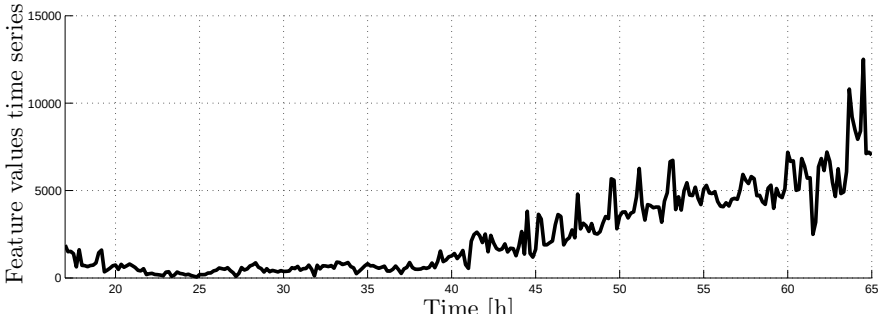


Figure 6.10: *vib*8 feature.

The results of the model estimation at different time steps are shown in Figure 6.11.

From Figure 6.11 it is clear, that up to the first 40 hours of operation the feature values are relatively small and there is no trend present. Consequently, the estimated system turns out to be stable (eigenvalue less than 1). When the feature values start to increase, the estimates of the system eigenvalues become greater than one. The increase in the system noise parameters corresponds to a stronger noise component present in the signal, which can be clearly seen from the time series (between 40 and 65 hours).

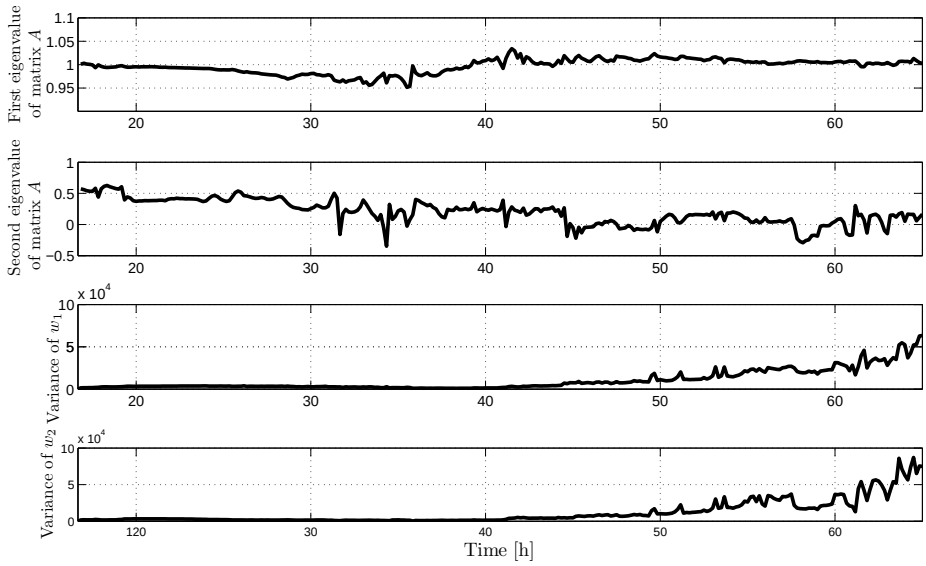


Figure 6.11: Values of the estimated parameters.

6.6.3 Predicting the distribution of the remaining useful life

Possessing the estimated stochastic dynamic model, described in the previous subsection, it is possible to predict the future behavior of the time sequences of the vibration feature.

Using the estimated linear model and Gaussian random variables, the system state can be propagated forward in time. The result is the predicted feature mean value and the variance of the estimate for every future time instance.

However, the goal of our work is not only to determine the future evolution of the feature value, but the distribution of the first passage time (FPT). Unfortunately, this cannot be calculated analytically.

The distribution of the FPT is therefore calculated through a Monte-Carlo simulation of the stochastic model in the following way. Assume we are at discrete time instance t. The simulation starts with the state vector $\mathbf{x}_t$, randomly sampled from the state distribution at time t $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$. This initial value is then propagated through the stochastic model, with a realization of the noise processes in the model. This is repeated for as long as the system output reaches the critical value. If the simulation is repeated enough times, the distribution of the times at which the simulated output reached the critical value can approximate the distribution of the RUL in terms of the FPT.

The critical values of the features (y^*) were set to values of 7000 and 1100 for *vib8* and *vib4*, respectively. The results on the time series from *vib8* are shown in Figure 6.12. Predictions of the RUL at three different time steps (45, 50 and 52) are given. The distribution of the predicted RUL is result of the Monte Carlo simulation with 500

repetitions.

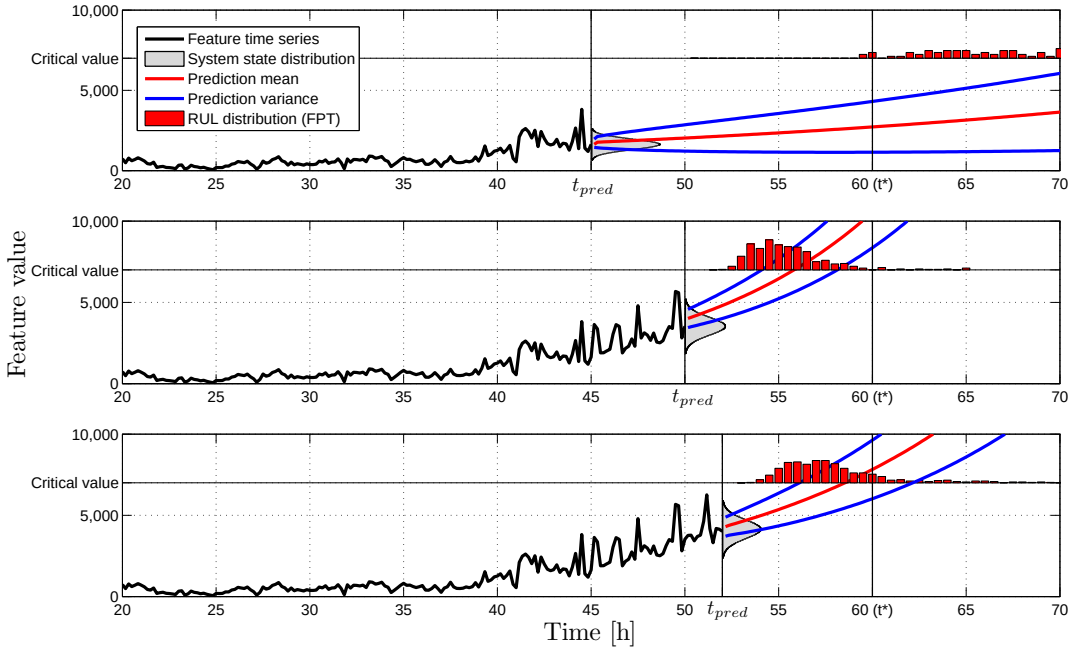


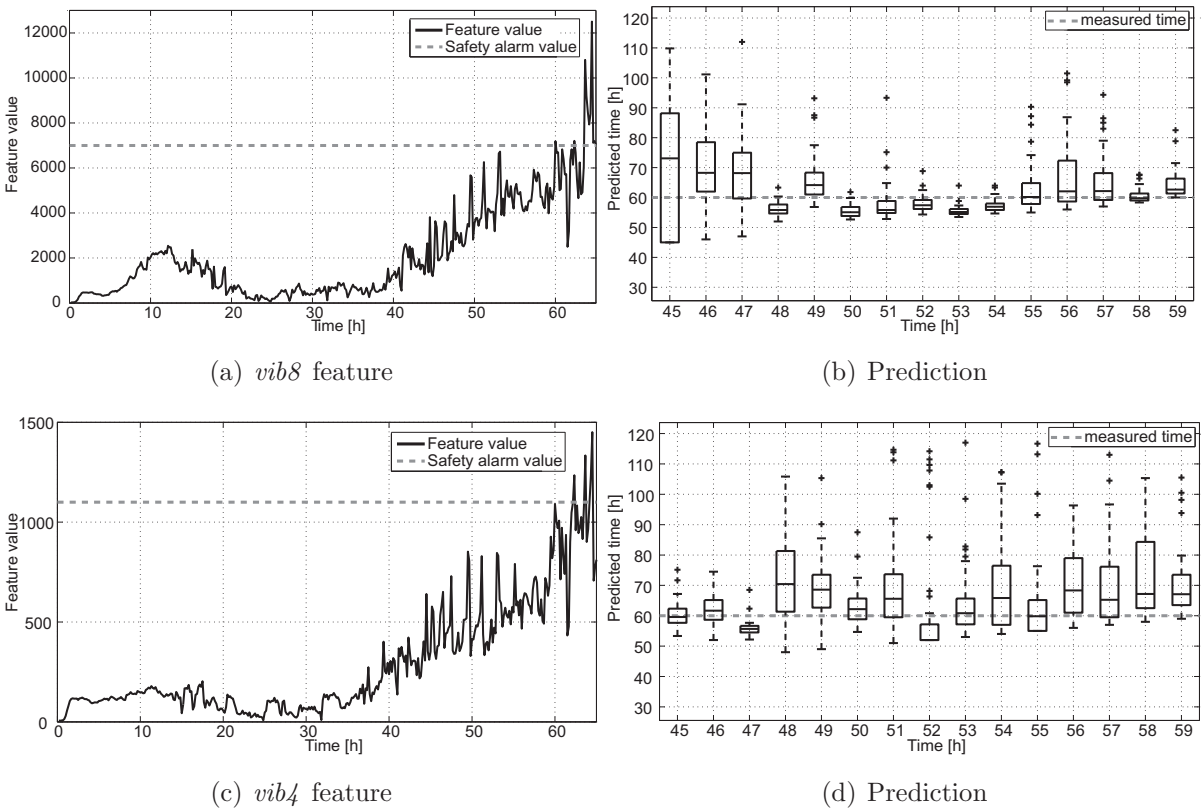
Figure 6.12: Estimation of distribution of the RUL at three different times ((a) 45 hours, (b) 50 hours and (c) 52 hours).

From Figure 6.12a it is clear, that at the time step of 45 hours, the algorithm detects a certain change in the underlying parameters, but the predicted distribution of the RUL is moved far into the future. As new measurements arrive and the trend in the feature values becomes more distinct (Figure 6.12b), the algorithm detects the change and adjusts the prediction accordingly. When moving closer to the actual first passage time ($t^* = 60$hours), the prediction becomes more accurate (Figure 6.12c).

To compare the results of the prediction at different time steps, the Monte Carlo simulation was repeated at 15 different times. The resulting distributions of the RUL are presented in Figure 6.13.

The initial non-monotonicity in the trend (the interval between 5 and 20 hours) is due to run-in process because the experiment has been started by a new pair of gears. This phenomena is not present in the model and the model prediction has only been validated on the feature values from $t = 20h$ onwards.

Since the distribution of the FPT is definitely not Gaussian and we cannot calculate it analytically, the uncertainty is quantified using a box plot. The representation is based on quartiles, where each of the four quartiles corresponds to 25% of the data points.



(a) *vib8* feature

(b) Prediction

(c) *vib4* feature

(d) Prediction

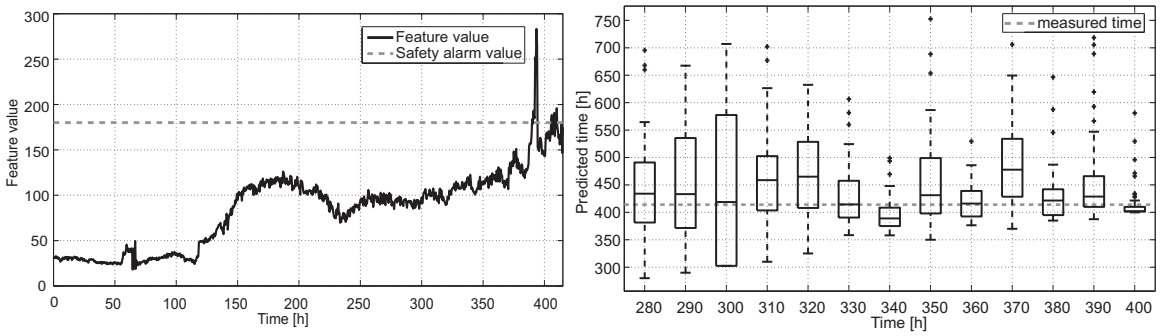
Figure 6.13: Results of the Monte Carlo analysis for the *vib8* ((a) and (b)) and *vib4* ((c) and (d)) sensor data.

Another appropriate method would be to provide the 95% confidence interval margins. But since the distribution is clearly not symmetrical, those results would not offer complete information.

It is clear, that the first reasonable estimates of the FPTs are made around 15 hours before the feature actually achieves the critical value. The estimation variance gradually decreases and approximately 10 hours before the actual FPT the uncertainty of the predicted FPT falls in the range ± 5 hours. However, due to the higher noise variance in *vib4* sensor data the variance of the predicted FPT is in this case (Figure 6.13 (c) and (d)) greater and is approximately ± 10 hours.

Finally, validation of the approach has been performed using data from a different experimental run using different quality of gears. The estimation procedure setup was the same as in the previous example. The only difference was, that because of the slower progression of the gear damage, the time window used was longer (500 samples). Results of the Monte Carlo analysis are given in Figure 6.14.

The feature time series shows slightly different behavior, compared to the previous experiment. This is due to the fact, that the validation experiment has been conducted



(a) Feature (b) Prediction

Figure 6.14: Results of the Monte Carlo analysis for the sensor data in validation experiment.

using a different set of gears. However, we can still make a relatively accurate prediction as early as 130 hours in advance.

From these results, we can conclude that the algorithm can successfully predict the time when the feature values reaches the critical level. Another important advantage of the algorithm lies in its simplicity in terms of the computational load. Both steps of the EM algorithms are based on the analytical solutions of filtering and optimization problems. The expressions that need to be calculated at these steps are all in the form of simple algebraic equations, thus consuming a minimal amount of computational time. Additionally, the number of iterations needed to satisfy the convergence criteria is relatively small, owing to the unimodality of the log-likelihood function of the linear state-space models. The low computational load of the algorithm may prove to be one of the key factors when considering the industrial applications of prognostics systems.

6.7 Nonlinear model of gear degradation process

After successful implementation of the linearized model with all advantages it provides, one naturally asks weather the use of nonlinear model of damage processes could bring additional benefits in terms of more accurate prediction.

The physics-based model for a system may be complex, but it is possible to represent the dynamics of gear wear using the much simpler population-growth-based model (Orchard and Vachtsevanos, 2009).

$$\begin{aligned} x_1(t+1) &= x_1(t) + a(b + cx_2(t))^2 + w_1(t) \\ x_2(t+1) &= x_2(t) + w_2(t) \\ y(t) &= x_1(t) + v(t) \end{aligned} \tag{6.11}$$

where the unknown model parameter vector is thus represented by $\boldsymbol{\theta} = [a, b, c, \mathbf{Q}, r]$. $\mathbf{Q}$ and r are the covariance matrices of process noise $\mathbf{w}(t)$ and measurement noise $v(t)$.

6.7.1 On line update of the model parameters

The model parameter values are updated every time a new feature value is obtained. This is done using the UTEM algorithm, as low computational load of the algorithm may prove to be one of thy key factors when considering industrial applications of the prognostics systems. In this case, the data used for the parameter estimation consists of the entire history of the vibration feature data. The model now serves as a complete description of the dynamics instead of a local approximation.

The UTEM algorithm is based on the implementation of Algorithm 4.2, where the UT scaling parameter is set to $\alpha = 0.7$. The initial guess of the model parameters is

$$\boldsymbol{\theta}^* = [\ a,\ b,\ c\ ] = [\ 3{\cdot}\,10^{-4},\ 100,\ 0.2\ ] \tag{6.12}$$

The noise covariance parameters were fixed to

$$\mathbf{Q} = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, \qquad r = \left[\ 10^3\ \right] \tag{6.13}$$

where the values of the covariance parameters were determined empirically. The estimated parameters at different time instances are shown in Figure 6.15.

As the damage progresses, the dynamics of the feature time series changes and the algorithm updates the parameter estimates accordingly. However, since the model is unstable, the parameter estimates are not consistent. The goal of this work is to obtain the best possible prediction, not to find the exact values of the parameters and this result should only be validated with respect to prediction quality.

6.7.2 Predicting of the distribution of the remaining useful life

Using the estimated nonlinear dynamic model, described in the previous subsection, it is possible to predict the future behavior of the time sequences of the vibration feature. The goal of the prediction is to determine the distribution of the times, at which the feature value will exceed the critical value (FPT).

The distribution of the FPT is calculated by Monte Carlo simulation of the stochastic model (6.11) in the similar way as in the linear case. At the time t_{pred}, at which we are making prediction, the simulation starts with initial value x_0, randomly selected from the initial state distribution $p(x_0)$. This initial value is then propagated trough the stochastic model, with certain realization of the stochastic processes in the model. This is repeated

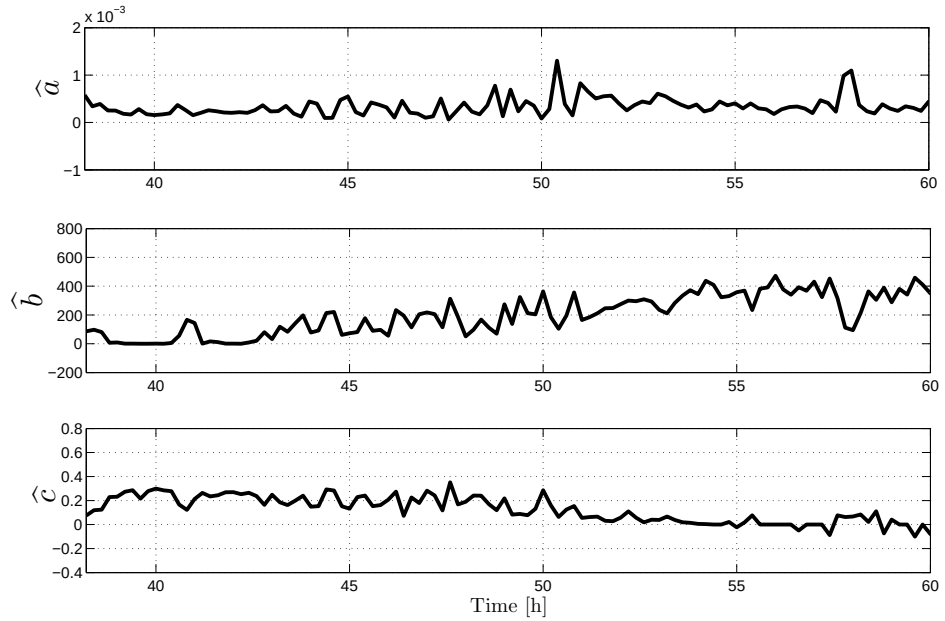


Figure 6.15: Values of the estimated model parameters.

until the system output reaches the critical value. The predicted probability density functions are shown in Figure 6.16 for predictions made at different times.

The goal of the predictions is to determine the distribution of the remaining useful life. To estimate it, the time at which the predicted feature value exceeds the critical value are determined for each run of the Monte Carlo simulation. To compare the results of prediction at different time steps, the Monte Carlo simulation has been performed for ten different values of T_{pred}. The resulting distributions of the RUL are presented in Figure 6.17. The actual feature value reached the critical value at time $t^* = 60h$.

It can be seen, that the first reasonable estimates of the RUL are made around 15 hours before the feature actually achieves the critical value. As we move closer to the actual time of failure, the variance of the distribution gradually decreases and the mean value is approaching the actual RUL. Apart from relatively accurate prediction, the important advantage of the algorithm lies in its simplicity in terms of computational load.

6.8 Summary of the experimental results

We presented a model-based approach for estimating the distribution of remaining useful life of a gear system. Two dynamic models are used to model the vibration signal time series. First, a linear model is considered as a local approximation of the feature time series and secondly, a nonlinear dynamic model is used.

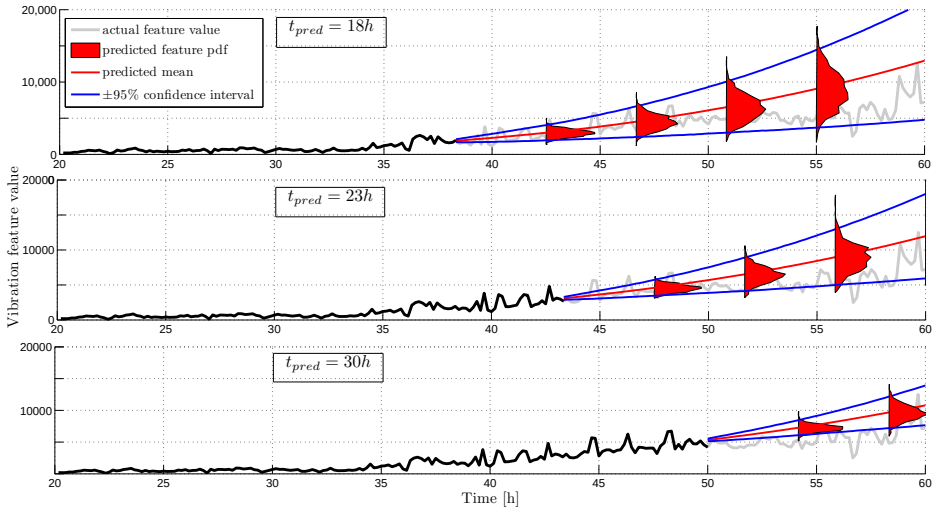
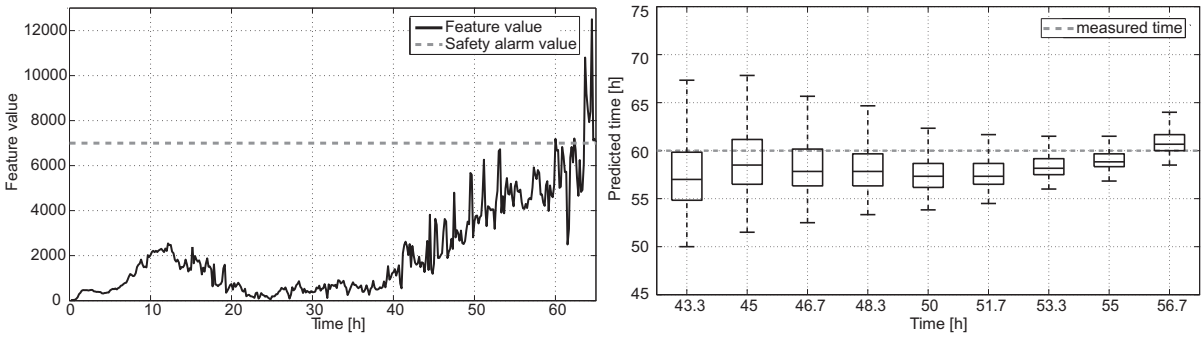


Figure 6.16: Predicted probability density functions using the estimated models at different time steps.



(a) *vib8* feature (b) Prediction

Figure 6.17: Results of the Monte Carlo analysis for the predicted EOL distributions ((a) - feature value, (b) - boxplot of the distributions).

The main advantage of the first approach is, that the use of linear model greatly simplifies parameter estimation. For the E-step of the expectation-maximization algorithm, the linear Kalman filtering theory can be used to estimate the distributions of hidden system state. Linear Kalman filter and RTS smoother algorithms can be used for its calculation. Furthermore, the log-likelihood function maximization problem can be solved analytically and the algorithm only involves calculation of relatively undemanding algebraic equations. Both steps thus use relatively little computational time. These steps must be repeated several times, before the estimate satisfies convergence criterion. The unimodality of the data log-likelihood function enables the algorithm to locate the function maximum in relatively few iterations (on average, 10 iteration in our case). Altogether, the computationally relatively undemanding calculation and fast convergence

make this algorithm attractive for deployment in industrial devices for online monitoring and prognostics.

Our experimental results show, that despite its simple structure the model obtained in this way can effectively predict the future behavior of the feature and can therefore be used to predict the distribution of the remaining useful life of the device. Based on the results of current experiments, it can be concluded that the accurate prediction can be made approximately 15 hours in advance. In the industrial applications, where there would be no accelerated damage mechanism and dynamics of the damage process are slower, the accurate estimates could probably be made even sooner. This offers the machine operators or maintenance staff a reasonable amount of time to replace the gear system without causing unnecessary production downtime.

In the second approach, the UTEM algorithm is used for on line parameter estimation of a nonlinear dynamic model. The advantage of the nonlinear model compared to the linear one is that it can be used to model the damage evolution in entire life cycle, thus achieving better quality prediction. The downside is, that the parameter estimation is more complex. The proposed implementation of UTEM algorithm greatly simplifies the computational complexity of the algorithm, while still achieving high accuracy of the estimates. The computationally relatively undemanding calculation and fast convergence make this algorithm attractive for deployment in industrial devices for on line monitoring and prognostics.

The results show, that this model can also effectively predict the future behavior of the feature and predict the distribution of the remaining useful life of the gears. However, there was no significant difference between the two models in terms of prediction accuracy. Partial explanation for this can be found by examining the estimated parameter values (c.f. Figure 6.15). Inserting their values in the model (6.11) results in a model with negligible nonlinearities and is. This motivates the search for a more informative nonlinear model of the degradation processes in the gears.

In this stage, our algorithms have only been tested in the case of constant load. In real application, it is common that the load as well as rounds per minute are changing in time. The key advantage of the model-based prognostics using dynamic models is also that it allows the modification of the algorithm to account for changing operating conditions. The main contribution of this work is thus not the only prediction quality, but the fact that it enables the prediction of the RUL under non-stationary operating conditions. The next step in the development of the procedure is to modify the algorithm so that it will include these as a measured system input and validate it with experimental data.

7 Conclusions and future work

This dissertation presented a new algorithm for system identification of state-space models, based on expectation-maximization algorithm. The algorithm was developed to solve the problem of joint estimation of model parameters and system states in nonlinear dynamical systems from observed input and output data. The theoretical background of state and parameter estimation of nonlinear systems was given, followed by the expectation-maximization algorithm for joint estimation. Based on that, a new algorithm was developed and evaluated using simulated numerical examples.

The expectation-maximization algorithm enables computation of the maximum likelihood estimates of the unknown parameters by iteratively maximizing the likelihood function with respect to the system states and unknown model parameters. The individual steps of the EM algorithm can be performed with different methods, depending on the properties of the underlying system. Nonlinear and non-Gaussian state-space models pose a demanding system identification problem, because closed-form solutions of the individual steps are not possible and one has to rely on the numerical approximations.

The main contribution of this dissertation is a new implementation of the expectation-maximization algorithm for nonlinear systems that is entirely based on the computationally efficient unscented transformation is presented. The unscented transformation provides the means of approximating probability density function using a set of deterministically chosen sigma-points with the corresponding weights. This method guarantees that the computational load of the algorithm is limited even when the dimension of the state-space increases.

With the novel UTEM algorithm we have successfully alleviated the problem related to the computational load of the approximation algorithms. The UTEM algorithm fills the gap between the closed form solutions that can be obtained in linear and Gaussian systems and the computationally complex sequential Monte Carlo based algorithm for nonlinear and non-Gaussian systems. Sequential Monte Carlo approach relies on a Dirac point mass approximation of the probability density function using a set of random particles. The approach results in the SMCEM algorithm, which can successfully approximate non-Gaussian distributions. However, its accuracy depends on the number of particles used and when the dimension of the state-space increases, the computation of the estimates becomes demanding.

The performance of the proposed algorithm was evaluated using numerical examples and we showed that the UTEM algorithm can successfully perform parameter and state estimation in nonlinear models. However, the approximation errors can in some cases preclude the algorithm from converging, which is even more evident when the unknown parameters are noise variances. According to our analysis, the SMCEM algorithm proved to be more stable in terms of convergence.

In the second part we presented a practical application of the expectation-maximization algorithm for system identification in the area of model-based prognostics of mechanical drives. The key is that gear damage evolution is modeled using a dynamic state-space models with unknown parameters. This is an important research result, as state-space formulation of the model is beneficial for two main reasons. First, by further investigation of the mechanical processes, additional knowledge of material degradation or measurement processes can be incorporated in the model, which can enable to link the state values to actual damage on the gear. Second possible advancement is, that by including the measured input (such as rotation speed or torque), the model will be able to predict the remaining useful life under non-stationary conditions.

The parameter values were estimated using the expectation-maximization algorithms for two different models of degradation process; linear and nonlinear. In both cases, the parameter values, estimated by the algorithm, provided a suitable model that can be used to predict the future evolution of the gear damage and thus enables the estimation of the component remaining useful life. The similar quality of prediction motivates the use of linear models especially when no detailed structure of the nonlinear model is known a priori. The on-line estimation of model parameters in this case is more stable and computationally more efficient. Nonlinear model could provide better quality of the estimates if the model was derived based on the equations of actual damage evolution and measurement process.

Future work in the direction of failure prognostics will proceed in two directions. First on is to determine a more exact nonlinear model structure, in which the system states will corresponds to the actual fault dimension and possibly include additional measure features (e.g. oil analysis). The second direction will be to develop this approach for non-stationary operating conditions. Preliminary results toward this goal have recently been submitted for publication, but still lack proper validation (Gašperin *et al.*, 2011b).

This dissertation demonstrated the importance of joint parameter and state estimation. The vast majority of the reported applications of state estimation assume that the model parameters are known in advance. A possible reason for this could be that the joint estimation proves to be a much more demanding problem and is therefore often intentionally ignored and only approximate values of the model parameters are used for

state estimation (this especially holds true for the noise covariances).

The Monte Carlo based approximation methods have attracted a lot of research attention in the past decade and important new results are constantly occurring. A more comprehensive comparison of the two approaches for system identification would certainly be interesting.

We have shown that in certain cases the UTEM algorithm estimates fail to converge. Unfortunately, the convergence conditions of this algorithm have not yet been established. Therefore, there is a strong need for a deeper understanding of the unscented transformation, especially in terms of quantifying or possibly decreasing the approximation error. Such a result would help to establish guidelines for when the UTEM algorithm can successfully perform its task and will therefore the main subject of our future work.

8 Acknowledgment

First of all I would like to express my gratitude toward my PhD supervisor, prof. dr. Đani Juričić for his guidance, support and encouragement in the last five years. Special thanks goes to the head of the Department of Systems and Control at Jožef Stefan Institute, Prof. Dr. Stanislav Strmčnik and former head deputy (now head) Dr. Vladimir Jovan for accepting me as an equal member of the group and providing a stimulating academic environment.

I also thank the members of my thesis committee: Prof. Dr. Stanislav Strmčnik, Prof. Dr. Juš Kocijan and Dr. Sc. Miroslav Kárný for their guidance, careful reading and commenting of my dissertation.

The past years were made a lot more enjoyable by all my friends at the Jožef Stefan Institute, Faculty of Electrical Engineering and Jožef Stefan International Postgraduate School, for which all of them deserve my gratitude. Special thanks among them goes to Pavle Boškoski, who always provided much needed comments on my work.

Finally, my greatest gratitude goes to my family, my mother and brother and most importantly to Nana for all her encouragement and support. Thank you for believing in me.

9 References

Alspach, D. and Sorenson, H. (1972). Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control*, **17(4)**, 438–448.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, **50**(1), 5–43.

Andrieu, C., Doucet, A., Singh, S. S., and Tadič, V. B. (2004). Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, **92**, 423–438.

Arulampalam, M. S., Maskell, S., and N. Gordon, T. C. (2002). A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, **50**, 174–186.

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

Bar-Shalom, Y., Li, R. X., and Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation*. Wiley Interscience, New Work, USA.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). Maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, **41**, 164–171.

Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *The Philosophical Transactions*, **53**, 370–418.

Bernouli, J. (1713). *Ars conjectandi, opus posthumum. Accedit Tractatus de seriebus infinitis, et epistola gallic scripta de ludo pilae reticularis*. Thurneysen Brothers, Basel.

Block, H. P. and Geitner, F. K. (1997). *Machinery Failure Analysis and Troubleshooting*. Gulf Publishing Co., Houston, TX, USA.

Borran, M. J. and Aazhang, B. (2002). EM-based multiuser detection in fast fading multipath environments. *In Proc. EURASIP*, **8**, 787–796.

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

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proceedings - Radar, Sonar and Navigation*, **146**(1), 2 –7.

Cempel, C. (1987). Simple condition forecasting techniques in vibroacoustical diagnostics. *Mechanical Systems and Signal Processing*, **1**(1), 75 – 82.

Chaari, F., Fakhfakh, T., and Haddar, M. (2009). Analytical modelling of spur gear tooth crack and influence on gearmesh stiffness. *European Journal of Mechanics - A/Solids*, **28**(3), 461 – 468.

Chen, Y. (2001). *Sequential importance sampling with resampling: Theory and Applications*. Ph.D. thesis, Stanford University.

Chitralekha, S. B., J.Prakash, H.Raghavan, R.B.Gopaluni, and S.L.Shah (2009). Comparison of expectation-maximization based parameter estimation using particle filter, unscented and extended kalman filtering techniques. In *Proceedings of the 15th IFAC Symposium on System Identification, July 6. - 8. 2009, Saint-Malo, France*.

Choy, F. K., Polyshchuk, V., Zakrajsek, J. J., Handschuh, R. F., and Townsend, D. P. (1996). Analysis of the effects of surface pitting and wear on the vibration of a gear transmission system. *Tribology International*, **29**(1), 77 – 83.

Combet, F. and Gelman, L. (2009). Optimal filtering of gearnext term signals for early damage detection based on the spectral kurtosis. *Mechanical Systems and Signal Processing*, **23**, 652–668.

Cramer, H. (1946). *Mathematical Methods of Statistics*. Princeton Univ. Press, Princeton, NJ, USA.

DeCastro, J. A., Liang, T., Kenneth, L. A., Goebel, K., and Vachtsevanos, G. (2009). Exact nonlinear filtering and prediction in process model-based prognostics. In *Proceedings of the 1st Annual conference of the PHM Society, September 27. - October 1., 2009, San Diego, USA*.

Dekking, F., Kraaikamp, C., Lopuhaa, H., and Meester, L. (2005). *A Modern Introduction to Probability and Statistics*. Springer-Verlag, London, UK.

Dempster, A., Lard, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, **39**, 1–38.

Deters, L. (2009). Tribology. In G. K.H and A. E.K., editors, *Handbook of Mechanical Engineering*. Springer, New York, USA.

Digalakis, V., Rohlicek, J. R., and Ostendorf, M. (1993). ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition. *IEEE Transactions on Speech and Audio Processing*, **1**(4), 431 –442.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, **10**, 1533–1375.

Doucet, A., Freitas, N. D., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, USA.

Durret, R. (1996). *Probability: Theory and Examples*. Cambridge University Press, Cambridge, UK.

Efron, B. (1982). *The Bootstrap, Jacknife and other Resampling Plans*. SIAM, Philadelphia, PA, USA.

Elliott, R. J., Aggoun, L., and Moore, J. B. (1995). *Hidden Markov Models: Estimation and Control*. Springer Science+Businedd Media, LLC, New York, USA.

Elsayed, A. E. (1996). *Reliability Engineering*. Prentice Hall, Upper Saddle River, USA.

Fajdiga, G., Flašker, J., Glodež, S., and Hellen, T. K. (2003). Numerical modelling of micro-pitting of gear teeth flanks. *Fatigue & Fracture of Engineering Materials & Structures*, **26**(12), 1135–1143.

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

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

Gauss, C. F. (1809). *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*.

Gašperin, M. and Juričić, D. (2011). Application of unscented transformation in nonlinear system identification. In *IFAC World Congress, August 28. - September 2., 2011, Milano, Italy (Accepted for publication)*.

Gašperin, M., Boškoski, P., and Juričić, D. (2009a). Gear health monitoring and prognosis. In *Proceedings of the 10th International PhD Workshop on Systems and Control, September 22. - 26., 2009, Hluboka nad Vitavou, Czech Republic*.

Gašperin, M., Boškoski, P., and Juričić, D. (2009b). Prognosis of gear health using stochastic dynamical models with online parameter estimation. In *Proceedings of the 1st Annual conference of the PHM Society, September 27. - October 1., 2009, San Diego, USA*.

Gašperin, M., Boškoski, P., and Juričić, D. (2009c). Prognosis of gear health using stochastic dynamical models with online parameter estimation. In *Proceedings of the 1st Annual conference of the PHM Society, September 27. - October 1., 2009, San Diego, USA*.

Gašperin, M., Juričić, D., Boškoski, P., and Vižintin, J. (2011a). Model-based prognostics of gear health using stochastic nonlinear dynamical models. *International Journal of Condition Monitoring*, **(To appear)**.

Gašperin, M., Boškoski, P., and Juričić, D. (2011b). Model-based prognostics under non-stationary operating conditions. In *Proceedings of Annual Conference of the Prognostics and Health Management Society 2011, September 25. - 29. 2011, Montreal, QC, Canada (Submitted)*.

Gibson, S. and Ninness, B. (2005). Robust maximum-likelihood estimation of multivariable dynamic systems. *Automatica*, **41**, 1667–1682.

Gilks, W., Richardson, S., and Spiegelhalter, D. (1996). *Markov chain Monte Carlo in practice*. Chapman & Hall, London, UK.

Goodwin, G. C. and Payne, R. L. (1977). *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press, New York, USA.

Gopaluni, R. B. (2008). Identification of nonlinear processes with known model structure under missing observatrions. In *Proceedings of the IFAC 17th World Congress, July 6.-11. 2008, Seoul, Korea*.

Gordon, N. J., Salmond, D. J., and 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.

Green, P. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **vol. 82**, 711 – 732.

Grenander, U. and Miller, M. (2007). *Pattern Theory: From Representation to Inference*. Oxford University Press, Oxford, UK.

Gut, A. (2009). *An Intermediate Course in Probability*. Springer Publishing Company, Incorporated, 2nd edition.

Hammersley, J. M. and Morton, K. W. (1954). Poor man's Monte Carlo. *Journal of the Royal Statistical Society, Series B*, **16(1)**, 23:38.

Haykin, S., editor (2001). *Kalman Filtering and Neural Networks*. John Wiley & Sons, New York, USA.

Heng, A., Zhang, S., Tan, A. C., and Mathew, J. (2009). Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing*, **23**, 724–739.

Higuchi, T. (1997). Monte Carlo filter using the genetic algorithm operators. *Journal of Statistical Computation and Simulation*, **59(1)**, 1 – 23.

Ho, D. and Randall, R. B. (2000). Optimisation of bearing diagnostic techniques using simulated and actual bearing fault signals. *Mechanical Systems and Signal Processing*, **14**, 763–788.

Hogben, L. (2006). *Handbook of Linear Algebra*. Chapman & Hall/CRC, Boka Raton, USA.

Hu, X., Schön, T. B., and Ljung, L. (2008). A basic convergence result for particle filtering. *IEEE Transactions on Signal Processing*, **56(4)**, 1337–1348.

Isard, M. and Blake, A. (1998). A smoothing filter for condensation. In *Proceedings of 5th European Conference on Computer Vision, volume 1, pages 767 - 781*.

ISO 10825 (1995). *Gears - Wear and damage to gear teet*. ISO.

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

Jensen, J. L. W. (1906). Sur les fonctions convexes et les ingalits entre les valeurs moyennes. *Acta Mathematica*, **30 (1)**, 175193.

Jordan, M., editor (1998). *Introduction to Monte Carlo methods*. Kluwer academic Publishers, Norwell, USA.

Julier, S. (2002). The scaled unscented transformation. In *Proceedings of the 2002 American Control Conference*.

Julier, S. and Uhlmann, J. (1996). A general method for approximating nonlinear transformations of probability distributions. Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, UK.

Julier, S. and Uhlmann, J. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, **92**(3), 401 – 422.

Julier, S. J. (1997). *Comprehensive Process Models for High-Speed Navigation*. Ph.D. thesis, University of Oxford, England.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of the Royal Statistical Society*, **82,(Series D)**, 35:45.

Kárný, M. (1996). Towards fully probabilistic control design. *Automatica*, **32**(12), 1719 – 1722.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, **5**(1), pp. 1–25.

Kotzalas, M. N. and Harris, T. A. (2001). Fatigue failure progression in ball bearings. *Journal of Tribology*, **123**(2), 238–242.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, **22(1)**, 79–86.

Lang, S. (1999). *Complex Analysis (Graduate Texts in Mathematics), 4th edition*. Springer Verlag, New York, USA.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer, New York, USA.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, **93(443)**, 1032–1044.

Ljung, L. (1996). Development of system identification.

Ljung, L. (1999). *System Identification: Theory for the User (2nd Edition)*. Prentice Hall, Upper Saddle River, N.J., USA.

Ljung, L. (2008). Perspectives on system identification. In *Proceedings of the 17th IFAC World Congress, Seoul, South Korea*.

Magnus, J. R. and Neudecker, H. (1999). *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons, Hoboken, New Jersy, USA, 2nd edition.

Marshall, A. (1959). The use of mult-stage sampling in schemes in monte carlo computations. In M. Meyer, editor, *Symposium on Monte Carlo Methods*. Wiley, New York, pp 123-140.

Maybeck, P. (1982). *Stochastic Models, Estimation and Control, Volume 2*. Academic Press, New York, USA.

McLachlan, G. J. and Krishnan, T. (2008). *The EM Algorithm and Extensions (2nd Edition)*. John Wiley and Sons, Hoboken, N. J., USA.

Metropolis, N. and Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, **44(247)**, 335:341.

M.S. Grewal, A. P. A. (2008). *Kalman Filtering: Theory and Practice Using MATLAB, 3rd Edition*. John Wiley & Sons, Hoboken, New Jersy, USA.

Musso, C., Oudjane, N., and Legland, F. (2001). Improving regularized particle filters. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice. New York*, number 12, pages 247–271. Statistics for Engineering and Information Science.

Newey, W. K. and McFadden, D. (1994). Large sample estimation and hypothesis testing. volume 4 of *Handbook of Econometrics*, pages 2111 – 2245. Elsevier, Amsterdam, Holland.

Ninnes, B. (2009). Some systems identification challenges and approaches. In *Proceedings of the 15th IFAC Symposium on System Identification, July 6. - 8. 2009, Saint-Malo, France*.

Norton, J. P. (2009). *An Introduction to Identification*. Dover Publications, Inc. New York, NY, USA.

Ødegaard, H., Rusten, B., and Westrum, T. (1994). A new moving bed biofilm reactor - applications and results. *Water Science and Technology*, **29(10-11)**, 157–165.

Øksendal, B. K. (2003). *Stochastic differential equations: an introduction with applications*. Springer-Verlag New York, LLC.

Orchard, M. E. and Vachtsevanos, G. J. (2009). A particle-filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*, **31**, 221–246.

Papoulis, A. (1984). *Probability, Random Variables, and Stochastic Processes, 2nd ed.* McGraw-Hill, New York, USA.

Penski, C. (2000). A new numerical method for SDEs and its application in circuit simulation. *Journal of Computational and Applied Mathematics*, **115**(1-2), 461 – 470.

Petersen, K. B. and Pedersen, M. S. (2008). The Matrix Cookbook.

Rauch, H. E., Tung, F., and Striebel, C. T. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, **3**, 1445–1450.

Robert, C. P. and Casella, G. (2005). *Monte Carlo Statistical Methods*. Springer-Verlag, New York, USA.

Rubin, D. B. (1988). Using the SIR algorithm to simulate posterior distributions. In M. H. Bernardo, K. M. Degroot, D. V. Lindley, and A. F. M. Smith, editors, *Bayesian Statistics 3*. Oxford University Press, Oxford, UK.

Särkkä, S. (2008). Unscented Rauch-Tung-Striebel smoother. *IEEE Transactions on Automatic Control*, **53**, 845–849.

Schein, O. and Denk, G. (1998). Numerical solution of stochastic differential-algebraic equations with applications to transient noise simulation of microelectronic circuits. *Journal of Computational and Applied Mathematics*, **100**(1), 77 – 92.

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

Schön, T. B. (2006). *Estimation of Nonlinear Dynamic Systems: Theory and Applications*. Ph.D. thesis, Linköpings University, Sweden.

Shumway, R. and Stoffer, D. (2005). *Time Series Analysis and Its Applications with R examples, Second Edition*. Springer, New York, USA.

Sikorska, J., Hodkiewicz, M., and Ma, L. (2010). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, **In Press**.

Simon, D. (2006). *Optimal state estimation: Kalman, H Infinity and Nonlinear Approaches*. John Wiley and Sons, Hoboken, N. J., USA.

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

Söderström, T. and Stoica, P. (1989). *System Identification*. Prentice Hall, Upper Saddle River, N.J., USA.

Stare, A., Vrečko, D., and Hvala, N. (2006). Modeling, identification, and validation of models for predictive ammonia control in a wastewater treatment plant - a case study. *ISA Transactions*, **45**, 159–174.

Suhač, B., Vižintin, J., Boškoski, P., and Juričić, D. (2008). Development of an intelligent rotating machinery diagnostics programme. In *The Fifth International Conference on Condition Monitoring and Machinery Failure Prevention Technologies, July 15. - 18. 2008, Edinburgh, UK*.

Suresh, S. (1998). *Fatigue of Materials 2nd edition*. Cambridge University Press, Cambridge, UK.

Tanizaki, H. (2001). Nonlinear and non-Gaussian state space modeling using sampling techniques. *Annals of the Institute of Statistical Mathematics*, **53(1)**, 63–81.

Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., and Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering systems*. Wiley, Hoboken, NJ, USA.

van de Geer, S. A. (2000). *Applications of Empirical Process Theory*. Cambridge University Press, Cambridge, UK.

van der Merwe, R. (2004). *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. Ph.D. thesis, Oregon Health & Science University, USA.

Verhaegen, M. and Verdult, V. (2007). *Filtering and System Identification*. Cambridge University Press, Cambridge, UK.

Wan, E. and van der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium, October 1. - 4. 2000, Lake Louise, Canada*.

Wan, E. A., Merwe, R. V. D., and Nelson, A. T. (2000). Dual estimation and the unscented transformation. In *in Neural Information Processing Systems*, pages 666–672. MIT Press.

Wang, P., Propes, N., Khiripet, N., Li, Y., and Vachtsevanos, G. (1999). An integrated approach to machine fault diagnosis. In *IEEE Annual Textile, Fiber and Film Industry Technical Conference, 1999*, page 7 pp.

Wang, W. Q., Golnaragh, M. F., and Ismail, F. (2004). Prognosis of machine health condition using neuro-fuzzy systems. *Mechanical Systems and Signal Processing*, **18**, 813–831.

Wang, Y. and Zhang, W. J. (1998). Stochastic vibration model of gear transmission systems considering speed-dependent random errors. *Nonlinear Dynamics*, **17**, 187–203.

Winkler, R. (2004). Stochastic differential algebraic equations of index 1 and applications in circuit simulation. *J. Comput. Appl. Math.*, **163**(2), 435–463.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann, San Francisco, USA.

Yang, X., Lv, J., and Feng, X. (2010). Particle swarm optimization particle filtering for dual estimation. In *Proceedings of the 2010 International Conference on Intelligent Control and Information Processing (ICICIP)*, pages 184 –187.

Yibo, C., Xiaopeng, X., Yan, L., and Tianhuai, D. (2010). A study of the correlation between gear wear and vibration. In *Advanced Tribology*, pages 273–277. Springer Berlin Heidelberg, Berlin, Germany.

Zadeh, L. A. (1956). On the identification problem. *IRE Transactions on Circuit Theory*, **3**, 277–281.

List of Figures

List of Tables

List of Algorithms

A Appendix

A.1 Chebyshev Inequality

Theorem A.1. Chebyshev inequality *Let x be a discrete random variable with expected value $\mu_x = E(x)$, and let $\epsilon > 0$ be any positive real number. Then*

$$P\left(|x - \mu| \geq \epsilon\right) \leq \frac{\mathrm{Var}(x)}{\epsilon^2} \tag{A.1}$$

Proof. Let $p(x)$ denote the probability density function of x. Then the probability that x differs form μ by at least ϵ is given by

$$P\left(|x - \mu| \geq \epsilon\right) = \int_{|x-\mu|\geq\epsilon} p(x)dx \tag{A.2}$$

We know that

$$\mathrm{Var}(x) = \int_x (x-\mu)^2 p(x)dx \geq \int_{|x-\mu|\geq\epsilon} (x-\mu)^2 p(x)dx \tag{A.3}$$

where the last inequality holds, since all the summands are positive and we have restricted the range of integration. But the last term can also be written as

$$\int_{|x-\mu|\geq\epsilon} (x-\mu)^2 p(x)dx = \int_{|x-\mu|\geq\epsilon} \epsilon^2 p(x)dx = \epsilon^2 \int_{|x-\mu|\geq\epsilon} p(x)dx = \epsilon^2 P\left(|x-\mu|\geq\epsilon\right) \tag{A.4}$$

Therefore,

$$P\left(|x - \mu| \geq \epsilon\right) \leq \frac{\mathrm{Var}(x)}{\epsilon^2} \tag{A.5}$$

□

A.2 The law of large numbers

The law of large numbers provides a theoretical background for a rather intuitive fact that the expected value of the random process can be estimated by averaging the samples from this process. This law was first proved by Swiss mathematician Jacob Bernoulli in his work *Ars Conjectandi* (Bernouli, 1713).

Theorem A.2. Law of large nubmers *Let $x_1, x_2, \ldots, x_n$ be a sequence independent and identically distributed random variables, with finite expected value $\mu = E(x_i)$ and finite variance $\sigma^2 = \mathrm{Var}(x_i)$. Additionally, let $\bar{x}_n = (x_1 + x_2 + \ldots + x_n)/n$ be the average and $\epsilon > 0$. Then*

$$\lim_{n\to\infty} P\left(|\bar{x}_n - \mu| > \epsilon\right) = 0 \tag{A.6}$$

Proof. From the definition, we can write

$$P\left(|\bar{x}_n - \mu| > \epsilon\right) = P\left(|\bar{x}_n - E(\bar{x}_n)| > \epsilon\right) \tag{A.7}$$

Applying Chabyshev inequality, we obtain

$$P\left(|\bar{x}_n - E(\bar{x}_n)| > \epsilon\right) \leq \frac{\mathrm{Var}(x)}{\epsilon^2} \tag{A.8}$$

which, using $\mathrm{Var}(\bar{x}_n) = \sigma^2/n$ translates to

$$P\left(|\bar{x}_n - E(\bar{x}_n)| > \epsilon\right) \leq \frac{\sigma^2}{n\epsilon^2} \tag{A.9}$$

As n goes to infinity, the right-hand side vanishes, no matter how small ϵ is, which concludes the proof. □

A.3 Cramér-Rao inequality

The Cramér-Rao inequality (Cramer, 1946) expresses the lower limit to the variance of the estimators of deterministic parameters. This limit is referred to as *Cramér-Rao lower bound.*

Theorem A.3. Cramér-Rao inequality *Let $\hat{\boldsymbol{\theta}}(\mathbf{x}_{1:n})$ be an estimator of $\boldsymbol{\theta}$, such that $E\left[\hat{\boldsymbol{\theta}}(\mathbf{x}_{1:n})\right] = \boldsymbol{\theta}_0$ and assume that $\mathbf{x}_{1:n}$ has a probability density function $p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n})$, then*

$$E\left[\hat{\boldsymbol{\theta}}(\mathbf{x}_{1:n}) - \boldsymbol{\theta}_0\right]\left[\hat{\boldsymbol{\theta}}(\mathbf{x}_{1:n}) - \boldsymbol{\theta}_0\right]^T \geq \mathcal{I}(\boldsymbol{\theta}_0)^{-1} \tag{A.10}$$

where

$$\begin{aligned}\mathcal{I}(\boldsymbol{\theta}_0) &= E\left[\frac{\partial}{\partial\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\mathbf{x}_{1:n})\right]\left[\frac{\partial}{\partial\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\mathbf{x}_{1:n})\right]^T\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \\ &= -E\left[\frac{\partial^2}{\partial\boldsymbol{\theta}^2}\log p_{\boldsymbol{\theta}}(\mathbf{x}_{1:n})\right]\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}\end{aligned} \tag{A.11}$$

The matrix $\mathcal{I}(\boldsymbol{\theta}_0)$ is known as the Fisher information matrix.

Proof. From assumption $E\left[\hat{\boldsymbol{\theta}}(\mathbf{x}_{1:n})\right] = \boldsymbol{\theta}_0$ it follows that

$$\boldsymbol{\theta}_0 = \int \hat{\boldsymbol{\theta}}(\mathbf{x}_n) p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n}) d\mathbf{x}_{1:n} \tag{A.12}$$

By the definition of $p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n})$, we can write

$$1 = \int p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n}) d\mathbf{x}_{1:n} \tag{A.13}$$

Differentiating expressions (A.12) and (A.13) with respect to $\boldsymbol{\theta}_0$ gives

$$\begin{aligned} \mathbf{I} &= \int \hat{\boldsymbol{\theta}}(\mathbf{x}_n) \left[\frac{\partial}{\partial \boldsymbol{\theta}_0} p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n})\right]^T d\mathbf{x}_{1:n} = \int \hat{\boldsymbol{\theta}}(\mathbf{x}_n) \left[\frac{\partial}{\partial \boldsymbol{\theta}_0} \log p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n})\right]^T p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n}) d\mathbf{x}_{1:n} \\ &= E\left[\hat{\boldsymbol{\theta}}(\mathbf{x}_n) \left[\frac{\partial}{\partial \boldsymbol{\theta}_0} \log p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n})\right]^T\right] \end{aligned} \tag{A.14}$$

$$\begin{aligned} \mathbf{0} &= \int \left[\frac{\partial}{\partial \boldsymbol{\theta}_0} \log p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n})\right]^T d\mathbf{x}_{1:n} = \int \left[\frac{\partial}{\partial \boldsymbol{\theta}_0} \log p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n})\right]^T p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n}) d\mathbf{x}_{1:n} \\ &= E\left[\frac{\partial}{\partial \boldsymbol{\theta}_0} \log p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n})\right]^T \end{aligned} \tag{A.15}$$

Multiplying (A.15) by $\boldsymbol{\theta}_0$ and subtracting from (A.14) gives

$$E\left[\hat{\boldsymbol{\theta}}(\mathbf{x}_n) - \boldsymbol{\theta}_0\right] \left[\frac{\partial}{\partial \boldsymbol{\theta}_0} \log p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n})\right]^T = \mathbf{I} \tag{A.16}$$

Introducing the notations

$$\boldsymbol{\alpha} = \hat{\boldsymbol{\theta}}(\mathbf{x}_n) - \boldsymbol{\theta}_0, \qquad \boldsymbol{\beta} = \frac{\partial}{\partial \boldsymbol{\theta}_0} \log p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n}) \tag{A.17}$$

we can write

$$E\begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}^T = \begin{bmatrix} E\left[\boldsymbol{\alpha}\boldsymbol{\alpha}^T\right] & \mathbf{I} \\ \mathbf{I} & E\left[\boldsymbol{\beta}\boldsymbol{\beta}^T\right] \end{bmatrix} \geq \mathbf{0} \tag{A.18}$$

Where positive semidefinitness follows by construction.

Lemma A.1. *Consider the symmetric matrix*

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \tag{A.19}$$

If $\mathbf{H} \geq 0$*, then (Hogben, 2006)*

$$\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T \geq 0 \tag{A.20}$$

Using Lemma A.1 gives

$$E\left[\boldsymbol{\alpha}\boldsymbol{\alpha}^T\right] \geq \left[E\left[\boldsymbol{\beta}\boldsymbol{\beta}^T\right]\right]^{-1} \tag{A.21}$$

Inserting the substitutions from (A.17) gives (A.10) and proves the inequality. Furthermore, differentiating the transpose of (A.15) results in

$$\mathbf{0} = \int \left[\frac{\partial^2}{\partial \boldsymbol{\theta}_0^2} \log p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n})\right] p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n}) d\mathbf{x}_{1:n} \tag{A.22}$$

$$+ \int \left[\frac{\partial}{\partial \boldsymbol{\theta}_0} \log p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n})\right] \left[\frac{\partial}{\partial \boldsymbol{\theta}_0} \log p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n})\right]^T p_{\boldsymbol{\theta}_0}(\mathbf{x}_{1:n}) d\mathbf{x}_{1:n} \tag{A.23}$$

which gives (A.11) and completes the proof. □

A.4 Jensen's inequality

Jensen's inequality (Jensen, 1906) describes relation between the value of a convex function of an integral to the integral of convex function (Durret, 1996).

Definition A.1. Convex function *Let f be a real valued function, defined on an interval $I = [a, b]$. f is said to be concave on I if $\forall x_1, x_2 \in I, \lambda \in [0, 1]$,*

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda) f(x_2) \tag{A.24}$$

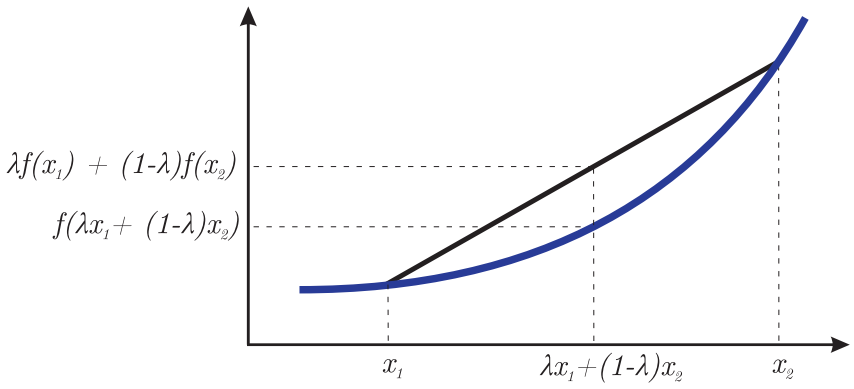


Figure A.1: Convex function f with secant line from x_1 to X_2

f is said to be strictly convex if the inequality is strict. This means, that the function falls above (strictly convex) or never below (convex) the straight line from points $(x_1, f(x_1))$ and $(x_2, f(x_2))$, c.f. Figure (A.1).

Theorem A.4. Jensen's inequality *Let $f(\cdot)$ be a convex function, defined on an interval I, furthermore $x_1, x_2, \ldots, x_n \in I$ and $\lambda_1, \lambda_2, \ldots, \lambda_n \geq 0$ with $\sum_{i=1}^{n} \lambda_i = 1$. Then*

$$\sum_{i=1}^{n} \lambda_i f(x_i) \geq f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \tag{A.25}$$

Proof. The finite form of Jensen's inequality can be proved by induction. For $n = 1$, the proof is trivial. For the case when $n = 2$, Equation A.25 corresponds to the definition of convexity (A.24). Using induction it can be shown, that this is also true for all natural numbers. Assume that the theorem is true for some n, then

$$\begin{aligned}
f\left(\sum_{i=1}^{n+1} \lambda_i x_i\right) &= f(\lambda_{n+1} x_{n+1} + \sum_{i=1}^{n} \lambda_i x_i) \\
&= f\left(\lambda_{n+1} x_{n+1} + (1 - \lambda_{n+1}) \frac{1}{1 - \lambda_{n+1}} \sum_{i=1}^{n} \lambda_i x_i\right) \\
&\leq \lambda_{n+1} f(x_{n+1}) + (1 - \lambda_{n+1}) f\left(\sum_{i=1}^{n} \frac{\lambda_i}{1 - \lambda_{n+1}} x_i\right) \\
&= \lambda_{n+1} f(x_{n+1}) + (1 - \lambda_{n+1}) f\left(\frac{1}{1 - \lambda_{n+1}} \sum_{i=1}^{n} \lambda_i x_i\right) \\
&\leq \lambda_{n+1} f(x_{n+1}) + (1 - \lambda_{n+1}) \sum_{i=1}^{n} \frac{\lambda_i}{1 - \lambda_{n+1}} f(x_i) \\
&= \lambda_{n+1} f(x_{n+1}) + \sum_{i=1}^{n} \lambda_i f(x_i) \\
&= \sum_{i=1}^{n+1} \lambda_i f(x_i)
\end{aligned} \tag{A.26}$$

□

In order to obtain the general inequality from this finite form, one needs to use a density argument. The finite form can be rewritten as:

$$\int f(x) d\mu_n(x) \geq f\left(\int x d\mu_n(x)\right) \tag{A.27}$$

where $\mu_n(x)$ is a measure given by Dirac's delta functions (δ)

$$\mu_n(x) = \sum_{i=1}^{n} \lambda_i \delta_{x_i} \tag{A.28}$$

Then, if $f(\cdot)$ is a convex function and $p(x)$ is a non-negative function such that $\int p(x)dx = 1$.

$$\int f(x) p(x) dx \geq f\left(\int x p(x) dx\right) \tag{A.29}$$

If $p(x)$ is a probability density function of x, this further simplifies to

$$E\left[f(x)\right] \geq f\left(E\left[x\right]\right) \tag{A.30}$$

A.5 Linear Kalman Filter

Theorem A.5. (Discrete-time Kalman filter)
Due to the properties of Gaussian distribution and its linear transformation, the resulting distributions are also Gaussian. They can be evaluated in closed form and the resulting distributions have the following form

$$p(\mathbf{x}_{t+1}|\mathbf{Y}_t) = \mathcal{N}(\mathbf{x}_{t+1}|\hat{\mathbf{x}}_{t+1|t},\ \mathbf{P}_{t+1|t}) \tag{A.31}$$
$$p(\mathbf{x}_{t+1}|\mathbf{Y}_{t+1}) = \mathcal{N}(\mathbf{x}_{t+1}|\hat{\mathbf{x}}_{t+1|t+1},\ \mathbf{P}_{t+1|t+1}) \tag{A.32}$$
$$p(\mathbf{y}_{t+1}|\mathbf{Y}_t) = \mathcal{N}(\mathbf{y}_{t+1}|\mathbf{C}\hat{\mathbf{x}}_{t+1|t},\ \mathbf{C}\mathbf{P}_{t+1|t}\mathbf{C}^T + \mathbf{R}) \tag{A.33}$$

Assuming that the initial state is distributed as $\mathbf{x}_0 \sim \mathcal{N}(\bar{\mathbf{x}}_0, \bar{\mathbf{P}}_0)$. *The parameters of the distributions can be computed with the Kalman filter* prediction *and* update *steps.*

$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{A}\hat{\mathbf{x}}_{t|t} + \mathbf{B}\mathbf{u}_t \tag{A.34a}$$
$$\mathbf{P}_{t+1|t} = \mathbf{A}\mathbf{P}_{t|t}\mathbf{A}^T + \mathbf{Q} \tag{A.34b}$$
$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1|t}\mathbf{C}^T(\mathbf{C}\mathbf{P}^-_{t+1|t}\mathbf{C}^T + \mathbf{R})^{-1} \tag{A.34c}$$
$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + \mathbf{K}_{t+1}[\mathbf{y}_{t+1} - \mathbf{C}\hat{\mathbf{x}}_{t+1|t} - \mathbf{D}\mathbf{u}_{t+1}] \tag{A.34d}$$
$$\mathbf{P}_{t+1|t+1} = \left(\mathbf{I} - \mathbf{K}_{t+1}\mathbf{C}\right)\mathbf{P}_{t+1|t} \tag{A.34e}$$

where $\hat{\mathbf{x}}_{0|0} = \bar{\mathbf{x}}_0$ *and* $\mathbf{P}_{0|0} = \bar{\mathbf{P}}_0$.

Proof. The joint distribution of $\mathbf{x}_{t+1}$ and $\mathbf{x}_t$ given $\mathbf{Y}_t$ is by Lemma 2.1 given as

$$\begin{aligned} p(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{Y}_t) &= p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_t) \\ &= \mathcal{N}(\mathbf{x}_{t+1}|\mathbf{A}_t\mathbf{x}_t + \mathbf{B}\mathbf{u}_t,\ \mathbf{Q})\mathcal{N}(\mathbf{x}_t|\hat{\mathbf{x}}_{t|t},\ \mathbf{P}_{t|t}) \\ &= \mathcal{N}\left(\left[\begin{array}{c}\mathbf{x}_t \\ \mathbf{x}_{t+1}\end{array}\right] \middle| \hat{\mathbf{x}}_1, \mathbf{P}_1\right) \end{aligned} \tag{A.35}$$

where

$$\hat{\mathbf{x}}_1 = \left[\begin{array}{c}\hat{\mathbf{x}}_{t|t} \\ \mathbf{A}\hat{\mathbf{x}}_{t|t} + \mathbf{B}\mathbf{u}_t\end{array}\right], \mathbf{P}_1 = \left[\begin{array}{cc}\mathbf{P}_{t|t} & \mathbf{P}_{t|t}\mathbf{A}^T \\ \mathbf{A}\mathbf{P}_{t|t} & \mathbf{A}\mathbf{P}_{t|t}\mathbf{A}^T + \mathbf{Q}\end{array}\right] \tag{A.36}$$

and the marginal distribution of $\mathbf{x}_{t+1}$ is by Lemma 2.2

$$p(\mathbf{x}_{t+1}|\mathbf{Y}_t) = \mathcal{N}(\mathbf{x}_{t+1}|\hat{\mathbf{x}}_{t+1|t},\ \mathbf{P}_{t+1|t}) \tag{A.37}$$

where

$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{A}\hat{\mathbf{x}}_{t|t} + \mathbf{B}\mathbf{u}_t \tag{A.38}$$

$$\mathbf{P}_{t+1|t} = \mathbf{A}\mathbf{P}_{t|t}\mathbf{A}^T + \mathbf{Q} \tag{A.39}$$

The joint distribution of $\mathbf{y}_t$ and $\mathbf{x}_t$ is by Lemma 2.1

$$\begin{aligned} p(\mathbf{y}_{t+1}, \mathbf{x}_{t+1}|\mathbf{Y}_t) =& p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{Y}_t) \\ =& \mathcal{N}(\mathbf{y}_{t+1}|\mathbf{C}\mathbf{x}_{t+1} + \mathbf{D}\mathbf{u}_{t+1},\ \mathbf{R})\mathcal{N}(\mathbf{x}_{t+1}|\hat{\mathbf{x}}_{t+1|t},\ \mathbf{P}_{t+1|t}) \\ =& \mathcal{N}\left(\left[\begin{array}{c}\mathbf{x}_{t+1}\\ \mathbf{y}_{t+1}\end{array}\right]\middle|\hat{\mathbf{x}}_2,\ \mathbf{P}_2\right) \end{aligned} \tag{A.40}$$

where

$$\hat{\mathbf{x}}_2 = \left[\begin{array}{c}\hat{\mathbf{x}}_{t+1|t}\\ \mathbf{C}\hat{\mathbf{x}}_{t+1|t} + \mathbf{D}\mathbf{u}_{t+1}\end{array}\right], \mathbf{P}_2 = \left[\begin{array}{cc}\mathbf{P}_{t+1|t} & \mathbf{P}_{t+1|t}\mathbf{C}^T\\ \mathbf{C}\mathbf{P}_{t+1|t} & \mathbf{C}\mathbf{P}_{t+1|t}\mathbf{C}^T + \mathbf{R}\end{array}\right] \tag{A.41}$$

And by Lemma 2.2, the conditional distribution of $\mathbf{x}_{t+1}$ is

$$\begin{aligned} p(\mathbf{x}_{t+1}|\mathbf{y}_{t+1}, \mathbf{Y}_t) =& p(\mathbf{x}_{t+1}|\mathbf{Y}_{t+1}) \\ =& N(\mathbf{x}_{t+1}|\hat{\mathbf{x}}_{t+1|t+1},\ \mathbf{P}_{t+1|t+1}) \end{aligned} \tag{A.42}$$

where

$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1|t}\mathbf{C}^T(\mathbf{C}\mathbf{P}^-_{t+1|t}\mathbf{C}^T + \mathbf{R})^{-1} \tag{A.43}$$

$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + \mathbf{K}_{t+1}\left[\mathbf{y}_{t+1} - \mathbf{C}\hat{\mathbf{x}}_{t+1|t} - \mathbf{D}\mathbf{u}_{t+1}\right] \tag{A.44}$$

$$\mathbf{P}_{t+1|t+1} = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{C})\,\mathbf{P}_{t+1|t} \tag{A.45}$$

The expressions (A.38),(A.39),(A.43),(A.44) and (A.45) constitute the Kalman filter. □

A.6 Linear Kalman Smoother

Theorem A.6. (Discrete-time Kalman smoother)
The for a fixed-interval the discrete-time Kalman smoother (the RTS smoother) is given by the following backward recursion equations

$$\mathbf{S}_t = \mathbf{P}_{t|t}\mathbf{A}_t^T\mathbf{P}^{-1}_{t+1|t} \tag{A.46a}$$

$$\hat{\mathbf{x}}_{t|T} = \hat{\mathbf{x}}_{t|t} + \mathbf{S}_t(\hat{\mathbf{x}}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) \tag{A.46b}$$

$$\mathbf{P}_{t|T} = \mathbf{P}_{t|t} + \mathbf{S}_t(\mathbf{P}_{t+1|T} - \mathbf{P}_{t+1|t})\mathbf{S}_t^T \tag{A.46c}$$

where $\hat{\mathbf{x}}_{t|t}$, $\hat{\mathbf{x}}_{t+1|t}$ and $\mathbf{P}_{t+1|t}$ are results from the Kalman filter algorithm and the smoother initial state at time T is also given by the Kalman filter as $\hat{\mathbf{x}}_{T|T}$ and $\mathbf{P}_{T|T}$.

Proof. The joint distribution of $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ given $\mathbf{Y}_t$ can be computed in the same manner as with the Kalman filter by Lemma 2.1.

$$\begin{aligned} p(\mathbf{x}_t, \mathbf{x}_{t+1}|\mathbf{Y}_t) &= p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_t) \\ &= \mathcal{N}(\mathbf{x}_{t+1}|\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t, \mathbf{Q})\mathcal{N}(\mathbf{x}_t|\hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}) \\ &= \mathcal{N}\left(\left[\begin{array}{c} \mathbf{x}_t \\ \mathbf{x}_{t+1} \end{array}\right] \middle| \hat{\mathbf{x}}_1, \mathbf{P}_1\right) \end{aligned} \tag{A.47}$$

where

$$\hat{\mathbf{x}}_1 = \left[\begin{array}{c} \hat{\mathbf{x}}_{t|t} \\ \mathbf{A}\hat{\mathbf{x}}_{t|t} + \mathbf{B}\mathbf{u}_t \end{array}\right], \mathbf{P}_1 = \left[\begin{array}{cc} \mathbf{P}_{t|t} & \mathbf{P}_{t|t}\mathbf{A}^T \\ \mathbf{A}\mathbf{P}_{t|t} & \mathbf{A}\mathbf{P}_{t|t}\mathbf{A}^T + \mathbf{Q} \end{array}\right] \tag{A.48}$$

Using the Markov property of the states

$$p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_T) = p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_t) \tag{A.49}$$

By Lemma 2.2, the conditional distribution is given by

$$p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_t) = N(\mathbf{x}_t|\hat{\mathbf{x}}_2,\ \mathbf{P}_2) \tag{A.50}$$

where

$$\mathbf{S}_t = \mathbf{P}_{t|t}\mathbf{A}^T\mathbf{P}_{t+1|t}^{-1} \tag{A.51}$$

$$\hat{\mathbf{x}}_2 = \hat{\mathbf{x}}_{t|t} + \mathbf{S}_t(\hat{\mathbf{x}}_{t+1|t+1} - \mathbf{A}\hat{\mathbf{x}}_{t|t} - \mathbf{B}\mathbf{u}_t) \tag{A.52}$$

$$\mathbf{P}_2 = \mathbf{P}_{t|t} - \mathbf{S}_t\mathbf{P}_{t+1|t}\mathbf{S}_t^{-1} \tag{A.53}$$

The joint distribution of $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ given $\mathbf{Y}_T$ is

$$\begin{aligned} p(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{Y}_T) &= p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{Y}_T)p(\mathbf{x}_{t+1}|\mathbf{Y}_T) \\ &= \mathcal{N}(\mathbf{x}_t|\hat{\mathbf{x}}_2,\ \mathbf{P}_2)\mathcal{N}(\mathbf{x}_{t+1}|\hat{\mathbf{x}}_{t+1|T},\ \mathbf{P}_{t+1|T}) \\ &= \mathcal{N}\left(\left[\begin{array}{c} \mathbf{x}_{t+1} \\ \mathbf{x}_t \end{array}\right] \middle| \hat{\mathbf{x}}_3,\ \mathbf{P}_3\right) \end{aligned} \tag{A.54}$$

where

$$\hat{\mathbf{x}}_3 = \left[\begin{array}{c} \hat{\mathbf{x}}_{t+1|T} \\ \hat{\mathbf{x}}_{t|t} + \mathbf{S}_t(\hat{\mathbf{x}}_{t+1|t+1} - \mathbf{A}\hat{\mathbf{x}}_{t|t} - \mathbf{B}\mathbf{u}_t) \end{array}\right], \mathbf{P}_3 = \left[\begin{array}{cc} \mathbf{P}_{t+1|T} & \mathbf{P}_{t+1|T}\mathbf{S}_t^T \\ \mathbf{S}_t\mathbf{P}_{t+1|T} & \mathbf{S}_k\mathbf{P}_{t+1|T}\mathbf{S}_t^T + \mathbf{P}_2 \end{array}\right] \tag{A.55}$$

Marginal distribution of $\mathbf{x}_t$ is given by Lemma 2.2 and is

$$p(\mathbf{x}_t|\mathbf{Y}_T) = \mathcal{N}(\mathbf{x}_t|\hat{\mathbf{x}}_{t|T},\ \mathbf{P}_{t|T}) \tag{A.56}$$

where expressions for $\hat{\mathbf{x}}_{t|T}$ and $\mathbf{P}_{t|T}$ simplify to

$$\hat{\mathbf{x}}_{t|T} = \hat{\mathbf{x}}_{t|t} + \mathbf{S}_t(\hat{\mathbf{x}}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) \tag{A.57}$$

$$\mathbf{P}_{t|T} = \mathbf{P}_{t|T} + \mathbf{S}_t(\mathbf{P}_{t+1|T} - \mathbf{P}_{t+1|t})\mathbf{S}_t^T \tag{A.58}$$

Expressions (A.51), (A.57) and (A.58) constitute the Kalman smoother algorithm and thus the proof is complete. □

B MATLAB source files

B.1 Unscented Transformation

```
%
% UT -  unscented transformation
%
% Syntax:  [mu,S,C,X,Y] = UT(M,P,g,alpha,beta,kappa)
%
% Inputs:   M: random variable mean
%           P: random variable covariance
%           g: transformation function handle
%           alpha: transformation parameter
%           beta: transformation parameter
%           kappa: transformation parameter
%
% Outputs:  mu: estimated mean
%           S: estimated covariance
%           C: estimated cross-covariance
%           X: sigma points of x
%           Y: sigma points of y
%

function [mu,S,C,X,Y] = ut(M,P,g,alpha,beta,kappa)

  % calculate UT weights
  [WM,WC,c] = ut_weights(size(M,1),alpha,beta,kappa);
  % form sigma-points
  X = ut_sigmas(M,P,c);
  % Propagate them through the function
  for i=1:size(X,2)
    Y = [Y feval(g,X(:,i))];
  end
  mu = zeros(size(Y,1),1);
  S  = zeros(size(Y,1),size(Y,1));
  C  = zeros(size(M,1),size(Y,1));
  for i=1:size(X,2)
    mu = mu + WM(i) * Y(:,i);
  end
  for i=1:size(X,2)
    S = S + WC(i) * (Y(:,i) - mu) * (Y(:,i) - mu)';
    C = C + WC(i) * (X(1:size(M,1),i) - M) * (Y(:,i) - mu)';
  end
end


function [WM,WC,c] = ut_weights(n,alpha,beta,kappa)
```

```
    % Compute the UT weights
    lambda = alpha^2 * (n + kappa) - n;
    WM = zeros(2*n+1,1);
    WC = zeros(2*n+1,1);
    for j=1:2*n+1
    if j==1
        wm = lambda / (n + lambda);
        wc = lambda / (n + lambda) + (1 - alpha^2 + beta);
    else
        wm = 1 / (2 * (n + lambda));
        wc = wm;
    end
        WM(j) = wm;
        WC(j) = wc;
    end
    c = n + lambda;
end

function X = ut_sigmas(M,P,c)

  A = chol(P)';
  X = [zeros(size(M)) A -A];
  X = sqrt(c)*X + repmat(M,1,size(X,2));
end
```

Listing B.1: Matlab code for Unscented Transformation.

B.2 Unscented Kalman Filter

```
%
%  UKF - Unscented Kalman Filter for nonlinear dynamic systems
%
% Syntax:  [XF,PF]=UTfilter(f,g,x0,P0,y,Q,R,alpha,beta,kappa)
%
% Inputs:   f: function handle for f(x)
%           g: fanction handle for g(x)
%           x0: initial state estimate
%           P0: initial state covariance
%           y: measurements
%           Q: process noise covariance
%           R: measurement noise covariance
%           alpha: transformation parameter
%           beta: transformation parameter
%           kappa: transformation parameter
%
% Output:   XF: filtered state estimate
%           PF: filtered state covariance
%
function [XF,PF]=UTfilter(f,g,x0,P0,y,Q,R,alpha,beta,kappa)

XF(:,1)=x0;
PF(:,:,1)=P0;
```

```
%Unscented Kalman filter
  for t=1:(length(y)-1)
    % Prediction step
    [X,P] = ut(XF(:,t),PF(:,:,t),f,alpha,beta,kappa);
    P = P + Q;
    % Update step
    [MU,S,C] = ut(X,P,g,alpha,beta,kappa);
    S = S + R;
    K = C / S;
    M = M + K * (Y(t) - MU);
    P = P - K * S * K';
    % Store estiamted state mean and covariance
    XF(:,t+1)=M;
    PF(:,:,t+1)=P;
  end
end
```

Listing B.2: Matlab code for Unscented Kalman Filter.

B.3 Unscented Kalman Smoother

```
% UTsmoother : Unscented Kalman Smoother
%
% Syntax:  [XS,PS,D] = UTsmoother(FX,FP,f,Q,alpha,beta,kappa)
%
% Input:   XF: NxK matrix of K mean estimates from Unscented Kalman filter
%          PF: NxNxK matrix of K state covariances from Unscented Kalman Filter
%          f: dynamic model function handle
%          Q: process noise covariance matrix
%          alpha: transformation parameter
%          beta: transformation parameter
%          kappa: transformation parameter
%
% Output:  XS: smoothed state mean
%          PS: smoothed state covariance
%          D: smoother gain

function [XS,PS,D] = UTsmoother(XF,PF,f,Q,alpha,beta,kappa)

% initialize the smooted mean and variance
% (note: different initialization is required if size(FX,1)==1)
XS(:,size(XF,2))=XF(:,size(XF,2));
PS(:,:,size(XF,2))=PF(:,:,size(XF,2));

  for k=(size(XF,2)-1):-1:1
    [XSp,PSp,C] = ut(XS(:,k),PS(:,:,k),f,alpha,beta,kappa);
    PS_pred = PS_pred + Q(:,:,k);
    D(:,:,k) = C/PSp;
    XS(:,k)   = XS(:,k) + D(:,:,k) * (XS(:,k+1) - XSp);
    PS(:,:,k) = PS(:,:,k) + D(:,:,k) * (PS(:,:,k+1) - PSp) * D(:,:,k)';
  end
end
```

Listing B.3: Matlab code for Unscented Kalman Smoother.

B.4 UT EM Algrotihm

```
% UT EM Algorithm : Unsceted Transform Expectation Maximization Algortihm
%
% Syntax:  [Q,param] = UT_EM_algortihm(param,y,x0,P0,f,g,alpha,beta,kappa)
%
% Input:   param: current parameter estimate
%          y: measurement vector
%          x0: initial state mean value
%          P0: initial state variance
%          f: process dynamics function handle
%          g: measurement function handle
%          alpha: UT parameter
%          beta: UT parameter
%          kappa: UT parameter
%
% Output:  Q: Q-function value
%          param: new parameter estimate
%

function [Q,param] = UT_EM_algortihm(param,y,x0,P0,f,g,alpha,beta,kappa)

% Run Kalman filter and smoother at current parameter estimate
[XF,PF]   = UTfilter(f,g,x0,P0,y,Q,R,alpha,beta,kappa,param);
[XS,PS,D] = UTsmoother(XF,PF,f,Q,alpha,beta,kappa,param);

% Compute sigma points for p(y_t|x_y) and p(x_t+1|x_t)
[xx, xx2] = sut_EM(XS,PS,D,PF,alpha,ka,y);

% Set optimizaiton parameters and start optimization
OPTIONS = optimset('MaxIter',100);
[param,Q] = fminsearch(@(param_o) Qfunc(param_o,f,g,xx,xx2,alpha,kappa,y),...
            param,OPTIONS);
end

function Q = Qfunc(param,f,g,xx,xx2,alpha,kappa,y)
% Optimization function
% Input:   param: current parameter estimate
%          f: process dynamics function handle
%          g: measurement function handle
%       xx: sigma point set for p(y_t|x_y)
%        xx2: sigma point set for  p(x_t+1|x_t)
%          alpha: UT parameter
%          kappa: UT parameter
%       y: measurement vector
%
% Output:  Q: Q-function value
%
```

```
  for j = 2 : length(xx)
    for i = 1:size(xx,2)
        I2t(i)=log(1/(sqrt(2*pi)*sqrt(det(R)))) + ((-1/2)* ...
                  (y(j)-h_func(xx(j,i),param) )' /R *...
                  (y(j)-h_func(xx(j,i),param) ) );
    end

    for i = 1:size(xx2,2)
        I1t(i)=log(1/(2*pi*sqrt(det(Q)))) + ((-1/2)* ...
                 (xx2(j,2,i)-f_func(xx2(j,1,i),param))' /Q *...
                 (xx2(j,2,i)-f_func(xx2(j,1,i),param)) );
    end
    L=1;
    lambda1 = alpha^2*(L+kappa)-L;
    L =2;
    lambda2 = alpha^2*(L+kappa)-L;

    Qt(j)=(lambda1/(1+lambda1))*I2t(1) ...
      + (1/(2*(lambda1+1)))*I2t(2)...
      + (1/(2*(lambda1+1)))*I2t(3)...
      + (lambda2/((lambda2+2)))*I1t(1)...
      + (1/(2*(lambda2+2)))*I1t(2)...
      + (1/(2*(lambda2+2)))*I1t(3)...
      + (1/(2*(lambda2+2)))*I1t(4)...
      + (1/(2*(lambda2+2)))*I1t(5)...
      ;
  end
  Q=-sum(Qt);

end

function [xx, xx2] = sut_EM(XS,PS,D,PF,alpha,kappa)
% Input:   XS: smoothed state mean
%          PS: smoothed state covariance
%          D: smoother gain
%          PF: filtered state covariance
%          alpha: UT parameter
%          kappa: UT parameter
%
% Output:  xx: sigma point set for p(y_t|x_y)
%         xx2: sigma point set for  p(x_t+1|x_t)
%
    xx = zeros(length(SX),1);
    xx2 = zeros(length(SX),2);
    % Form sigma-points to approximate p(y_t|x_t)
    L=1;
    lambda = alpha^2*(L+kappa)-L;
    for j=1:length(XS)
        A = squeeze(PS(:,:,j));
        rt = [zeros(size(XS(j))) chol((L+lambda)*A) -chol((L+lambda)*A)];
        rt = rt + repmat(XS(j),1,size(rt,2));
        xx(j,:)=rt;
    end
    % Form sigma-points to approximate p(x_t+1|x_t)
    L =2;
    lambda = alpha^2*(L+kappa)-L;
    for j=2:length(XS)
```

```
        M= [XS(j-1); XS(j)];
        A=[D(:,j-1)*PS(:,j)*D(:,j-1)'+PF(:,j), D(:,j-1)*PS(:,j);...
           D(:,j-1)*PS(:,j) ,                  PS(:,j)];
        rt = [zeros(size(M)) sqrtm((L+lambda)*A) -sqrtm((L+lambda)*A)];
        rt = rt + repmat(M,1,size(rt,2));
        xx2(j,:)=rt;
    end
end
```

Listing B.4: Matlab code UTEM algorithm.

C Publications related to this dissertation

The scientific contributions of this dissertations were published in the following papers:

Journal papers:

- Gašperin, M., Juričić, Đ., Boškoski, P., and Vižintin, J. (2011). Model-based prognostics of gear health using stochastic dynamical models. *Mechanical Systems and Signal Processing*, **25**(2), 537-548.

- Gašperin, M., Juričić, Đ., Boškoski, P., and Vižintin, J. (2011). Model-based prognostics of gear health using stochastic nonlinear dynamic models. *International Journal of Condition Monitoring.* (Submitted)

Conference papers:

- Gašperin, M., Boškoski, P., and Juričić, Đ. (2011). Model-based prognostics under non-stationary operating conditions. In *Proceedings of Annual Conference of the Prognostics and Health Management Society 2011, September 25. - 29. 2011, Montreal, Canada (Submitted).*

- Gašperin, M. and Juričić, Đ. (2011). Application of unscented transformation in nonlinear system identification. In *18th IFAC World Congress, August 28. - September 2. 2011, Milano, Italy (Accepted for publication).*

- Gašperin, M., Vrečko, D., and Juričić, Đ. (2010d). System identification of nonlinear dynamic models: Application to wastewater treatment plant. In *2010 IEEE Multiconference on System and Control : proceedings of the 19th International Conference on Control Applications, 10th IEEE International Symposium on Computer-Aided Control System Design (CACSD), 25th IEEE International Symposium on Intelligent Control (ISIC), September 8. - 10. 2010, Yokohama, Kanagawa, Japan.*

- Gašperin, M., Juričić, Đ., Boškoski, P., and Vižintin, J. (2010). Model-based prognostics of gear health using stochastic nonlinear dynamic models. In *The 7th In-*

ternational Conference on Condition Monitoring and Machinery Failure Prevention Technologies, June 22. - 24. 2010, Stratford-upon-Avon, England. .

- Gašperin, M., Juričić, Đ., and Boškoski, P. (2010). Condition prognosis of mechanical drives based on nonlinear dynamic models. In *Proceedings of Conference on Control and Fault-Tolerant Systems, SysTol'10, October 6. - 8. 2010, Nice, France* .

- Gašperin, M. and Juričić, Đ. (2010). Model-based prognostics of mechanical drives: the maximum-likelihood approach. In *Zbornik devetnajste mednarodne Elektrotehnike in raunalnike konference ERK 2010, September 20.- 22. 2010, Portorož, Slovenija*.

- Gašperin, M., Boškoski, P., and Juričić, Đ. (2009). Gear health monitoring and prognosis. In *Proceedings of the 10th International PhD Workshop on Systems and Control, September 22. - 26. 2009, Hluboka and Vitavou, Czech Republic*.

- Gašperin, M., Boškoski, P., and Juričić, Đ. (2009). Prognosis of gear health using stochastic dynamic models with online parameter estimation. In *Proceedings of Annual Conference of the Prognostics and Health Management Society 2009, September 27. - October 1. 2009, San Diego, USA*.