Linköping Studies in Science and Technology. Dissertations.
No. 924

Particle Filtering

for Positioning and Tracking Applications

Rickard Karlsson

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

Linköping 2005

Particle Filtering for Positioning and Tracking Applications

rickard@isy.liu.se
www.control.isy.liu.se
Department of Electrical Engineering
Linköping University
SE–581 83 Linköping
Sweden

ISBN 91-85297-34-8 ISSN 0345-7524

Printed in Sweden by UniTryck, Linköping, Sweden.

To Martina and Karin

Abstract

A Bayesian approach to positioning and tracking applications naturally leads to a recursive estimation formulation. The recently invented particle filter provides a numerical solution to the non-tractable recursive Bayesian estimation problem. As an alternative, traditional methods such as the extended Kalman filter, which is based on a linearized model and an assumption on Gaussian noise, yield approximate solutions.

In many practical applications, signal quantization and algorithmic complexity are fundamental issues. For measurement quantization, estimation performance is analyzed in detail. The algorithmic complexity is addressed for the marginalized particle filter, where the Kalman filter solves a linear subsystem subject to Gaussian noise efficiently.

The particle filter is adopted to several positioning and tracking applications and compared to traditional approaches. Particularly, the use of external database information to enhance estimation performance is discussed. In parallel, fundamental limits are derived analytically or numerically using the Cramér-Rao lower bound, and the result from estimation studies is compared to the corresponding lower bound. A framework for map-aided positioning at sea is developed, featuring an underwater positioning system using depth information and readings from a sonar sensor and a novel surface navigation system using radar measurements and sea chart information. Bayesian estimation techniques are also used to improve position accuracy for an industrial robot. The bearings-only tracking problem is addressed using Bayesian techniques and map information is used to improve the estimation performance. For multiple-target tracking problems data association is an important issue. A method to incorporate classical association methods when the estimation is based on the particle filter is presented. A real-time implementation of the particle filter as well as hypothesis testing is introduced for a collision avoidance application.

Acknowledgments

First, I would like to thank my supervisor, Professor Fredrik Gustafsson. Without his skillful guidance and intuition this thesis would not exist. Dr. Niclas Bergman is gratefully acknowledged for introducing me to the field of particle filtering and all excellent initial help. I would also like to thank Professor Lennart Ljung for accepting me in the control group and for his excellent management and support when needed. Moreover, Ulla Salaneck deserves extra gratitude for all administrative help and assistance, always with a cheerful attitude. The members and staff at the control group are recognized for creating such a stimulating and joyful environment. In particular Dr. Fredrik Tjärnström and Dr. Mikael Norrlöf deserve many thanks for all their good advice.

My co-authors; Professor Fredrik Gustafsson, Dr. Niclas Bergman, Dr. Mikael Norrlöf, Thomas Schön and Jonas Jansson are gratefully acknowledged for all stimulating discussions and research cooperation, that in the end resulted in various publications.

Several people have improved the quality of this thesis, for which I am very grateful. Gustaf Hendeby has proof-read most of the material and given valuable comments, and also spent a lot of time assisting with LaTeX formatting. Also Måns Östring has given valuable programming help during these years. Dr. Jonas Elbornsson has been of great help in proof-reading and discussing quantization issues. In addition, I would like to thank the following colleagues for proof-reading various parts of the thesis and for giving valuable comments and suggestions; Thomas Schön, Erik Wernholt, Dr. Mikael Norrlöf, Jonas Jansson and Dr. Ola Härkegård.

I would like to thank Saab Bofors Underwater Systems; Per-Ola Svensson and Elias Fransson for providing sonar data from a torpedo system, and Björn Johansson, Anna Falkenberg and Tobias Karlsson for providing underwater terrain information. Björn Gabrielsson provided assistance and fruitful discussions in underwater navigation. Erik Svensson helped with sea chart transformations. Thanks also to Volvo Car Corporation; Fredrik Lundholm and Lena Westervall, for providing measurement data and hardware to test the developed collision mitigation algorithm on.

I would like to thank my employer Saab Bofors Dynamics for encouraging me to engage in graduate studies and for allowing crucial leave of absence during my thesis work. The financial support from the VINNOVA Center of Excellence ISIS (Information Systems for Industrial Control and Supervision) at Linköping University, is gratefully acknowledged.

I would like to thank my parents, my sister, and my parents-in-law for all support during these years, and for baby-sitting when the deadline was approaching. All friends that have been neglected and that are still there are gratefully appreciated. In particular, I thank Sofia Petersson and Åsa Svensson for helping with baby-sitting at several occasions. Finally, my wife Karin for love and encouragement and our daughter Martina for keeping hard constraints on my working day and for always making us smile.

Linköping, January 2005

Rickard Karlsson

Contents

Notation

Symbols

C_R	Coefficient of variation
Δ	Quantization step or grid-size
δ	Delta-Dirac function (continuous or discrete)
e_t	Measurement noise
ϵ	Residual vector, innovations
f	State equation transition mapping (discrete-time)
F	Linearized state update matrix
$\gamma_t^{(i)}$	Weight for particle or model i
h	Measurement relation
H	Linearized measurement relation matrix
$\mathcal{I}$	Identity matrix
J	Fisher information
K	Kalman gain
ℓ	Likelihood function
$\mathcal{M}_t^{(i)}$	Model i at time t
N	Number of particles, samples or models
N_{eff}	Effective number of samples
N_F	Number of RPEKF filters
$\mathcal{N}(\mu, P)$	Gaussian distribution with mean value μ and covariance P
$\mathcal{O}$	Null matrix or ordo operator
P	Covariance matrix
$p(\cdot)$	Probability density function
$p(x_t\|\mathbb{Y}_t)$	Posterior density
p_{e_t}	Measurement noise probability density
p_{w_t}	Process noise probability density
p_{x_0}	Initial state probability density
Q	Process noise covariance matrix
R	Measurement noise covariance matrix
$\mathbb{R}^n$	Euclidean n-dimensional space
$q(\cdot)$	Proposal density
q	Scalar process noise

σ	Standard deviation
T	Sample period
$\mathcal{U}$	Uniform distribution
u	Input signal
w_t	Process noise
x_t	State vector at time t
$\mathbb{X}_t$	The set of state vectors: $\mathbb{X}_t = \{x_0, x_1, \ldots, x_t\}$
$\hat{x}_{t\|t}$	Estimate (filtering) at time t
$\hat{x}_{t\|t-1}$	Estimate (one step prediction) at time t
$x_t^{(i)}$	Sample or multiple-model selection of state vector
X, Y, Z	Cartesian position variables
y_t	Measurement at time t
$\mathbb{Y}_t$	The set of ordered measurements: $\mathbb{Y}_t = \{y_1, y_2, \ldots, y_t\} = \{y_i\}_{i=1}^t$

Operators and Functions

A^T	The transpose of a matrix A
$\arg\max_x$	The argument x that maximizes the operand
Cov	Covariance operator
$\mathbb{E}$	Expected mean
$\mathcal{F}$	Fourier transform
$\boldsymbol{\Delta}$	Laplacian operator (definition, see (2.16))
∇	Gradient or Jacobian operator (definition, see (2.16))
$\lfloor \cdot \rfloor$	Round downwards to the nearest integer
Prob	Probability
$\mathcal{Q}$	Quantization operator
$\succeq$	Positive semi-definte
$<u, v>$	Scalar product
Var	Variance

Abbreviations

APF	Auxiliary Particle Filter
AFMM	Adaptive Forgetting through Multiple Models
ATC	Air Traffic Control
AUN	Additive Uniform Noise
CF	Characteristic Function
CMbBS	Collision Mitigation by Braking System
CRLB	Cramér-Rao Lower Bound
DH	Denavit-Hartenberg
DSP	Digital Signal Processor
EKF	Extended Kalman Filter
EF	Equivalent Flops
FCAS	Forward Collision Avoidance System
flops	Floating-Point Operations

FIM	Fisher Information Matrix
FT	Fourier Transform
GPB	Generalized Pseudo-Bayesian
GPS	Global Position System
GS	Gaussian Sum
GPF	Gaussian Particle Filter
GSPF	Gaussian Sum Particle Filter
HMM	Hidden Markov Model
i.i.d.	Independent Identically Distributed
IMM	Interacting Multiple Model
INS	Inertial Navigation Systems
IR	Infrared
IS	Importance Sampling
JPDA	Joint Probabilistic Data Association
KF	Kalman Filter
LOS	Line-of-Sight
LS	Least Squares
MAP	Maximum a Posteriori
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
MHT	Multiple Hypotheses Tracking
ML	Maximum Likelihood
MMS	Minimum Mean Square
MPF	Marginalized Particle Filter
MSE	Mean Square Error
NN	Nearest Neighbor
pdf	Probability Density Function
PGT	Path Generation Toolbox
PF	Particle Filter
RMSE	Root Mean Square Error
RPEKF	Range Parameterized Extended Kalman Filter
RPF	Regularized Particle Filter
RTW	Real-Time Workshop
SIR	Sampling Importance Resampling
SIS	Sequential Importance Sampling
SNR	Signal to Noise Ratio
TWS	Track While Scan
UKF	Unscented Kalman Filter
UPF	Unscented Particle Filter
UW	Underwater

Part I

1

Introduction

In this thesis, several *positioning* and *target tracking* applications are discussed. Positioning is basically the concept of finding the own platform's kinematic state, i.e., position and velocity, whereas target tracking refers to the state of an observed object. In Figure 1.1, an Air-to-Sea scenario is depicted, where the tracking platform is an aircraft and the target a ship. In order to find the aircraft's kinematic state,

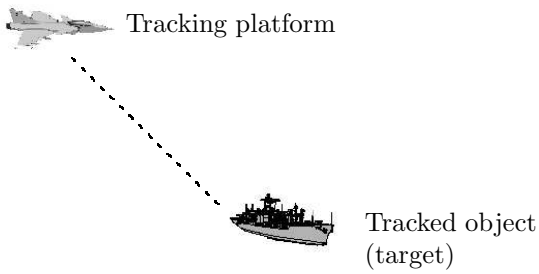


Figure 1.1: *An Air-to-Sea tracking application. The tracking platform is equipped with a positioning system, where the relative target range is measured, for instance using a radar sensor.*

an accurate positioning or navigation system is used. Usually, inertial sensors such as gyros and accelerometers are used. Another important source of information is the *global positioning system* (GPS). To find the target's position, measurements from tracking sensors are used. The aircraft can for instance be equipped with a radar sensor, measuring relative range to the target. Another common sensor is an *infrared* (IR) vision device, which is a passive sensor, so in principle only the direction not the range to the target is measured.

Usually, the sensor measures only part of the quantities that are of interest. Hence, there is a need to estimate the part not directly measured. Estimation methods are also used in order to decrease the influence of noise. In many practical applications there is a need for on-line computation. Therefore, recursive estimation methods are used. Depending on the method chosen, issues such as estimation performance and real-time aspects are important.

In Section 1.1, the main problems discussed in the thesis are defined and the problem formulation is given in Section 1.2. In Section 1.3, an outline of the thesis is given. A summary of the contributions is presented in Section 1.4.

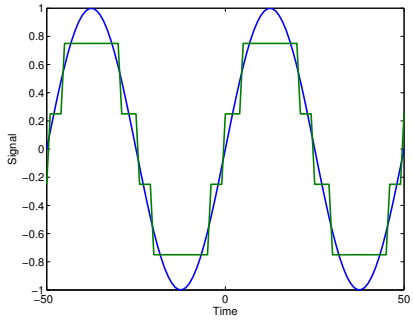
1.1 Topics

In this section, the main problem areas discussed in the thesis are introduced with references to individual papers in Part II.

1.1.1 Estimation Theory and Analysis

In practice, many estimation problems rely on on-line solutions, hence recursive methods are necessary. Traditional estimation methods for linear systems subject to Gaussian noise are based on the *Kalman filter* (KF), [47, 48] or for nonlinear systems the *extended Kalman filter* (EKF), [4, 37, 47]. In recent years, the growth in computational power has made computer intensive statistical methods feasible. Instead of approximating the system using linearization, the estimation problem is solved directly using Monte Carlo techniques, [26]. The main breakthrough came with the seminal paper on *particle filtering*, [36]. These methods are described in Chapters 2–3. In this thesis the particle filter will be analyzed and applied frequently, where the EKF is used as a comparison.

Estimation methods are often implemented in dedicated hardware in *digital signal processors* (DSPs) or on a desktop computer. Often, the resolution in the calculations is sufficient, so quantization effects can be neglected. However, the sensor may in many cases produce data that is quantized. Many sensors deliver data that is naturally quantized such as radar range, vision devices, and cogged wheels to measure angular speeds etc. Since cheap low-quality sensors have appeared on the market and in many consumer products, quantization effects may be an issue in many applications. Also, the increased use of distributed sensors in communication networks with limited bandwidth illustrates the need for many low-quality sensors. This motivates the study of how quantization affects estimation performance. In



(a) Signal quantization.

Instruction	Flops
$P_A := P_{t\|t}(A_t)^T$	$pk^2 + (k-1)kp$
$M := A_t P_A + Q_t$	$kp^2 + (k-1)p^2 + p^2$
$\vdots$	$\vdots$
$T_5 := F_t P_{t\|t}(F_t)^T$	$2k^3 + 2(k-1)k^2$

(b) Algorithmic computation complexity using floating-point operations (flops), where p and k are matrix dimensions.

Figure 1.2: *Two theoretical aspects: signal quantization and computational complexity of an algorithm by computing the number of operations.*

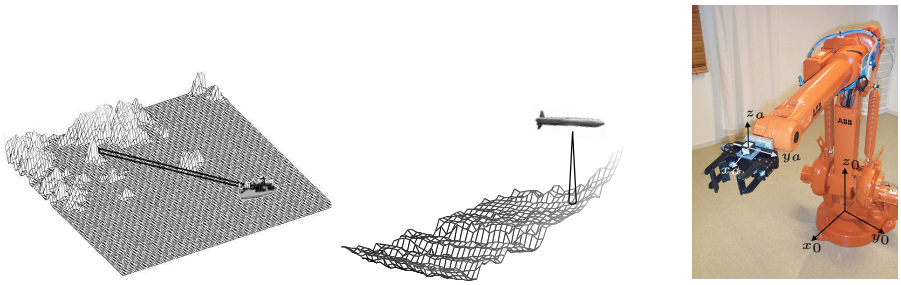
Figure 1.2 (a), signal quantization is depicted. In [103, 104], these effects are studied, where an analogy with sampling theory is developed for quantization. Measurement quantization for filtering and estimation is described in Paper A.

In many practical applications, real-time performance is important. Hence, efficient algorithms for recursive estimation are essential. In dedicated hardware, traditional algorithms can be analyzed by counting the number of *floating-point operations* (flops) for each instruction in the algorithms. In Figure 1.2 (b), an example of an algorithm and the number of flops for each code line is given. When computer intensive algorithms are used, new issues arise, since part of the computational complexity is related to other aspects than just basic operations. In Paper B, the algorithmic computational complexity of a particle filtering algorithm is analyzed.

An important aspect of any estimation method is its estimation performance. This can be analyzed using the *Cramér-Rao lower bound* (CRLB), [21, 66, 74], analytically (if possible) or numerically. In simulation studies, it can also be compared to the actual performance in terms of the *root mean square error* (RMSE).

1.1.2 Positioning

The concept of finding the own platform's (aircraft, ship, car, etc.) kinematic state, i.e., position and velocity is often referred to as positioning. This can be based on information from inertial sensors, such as gyros and accelerometers using dead-reckoning. The *global positioning system* (GPS) is an alternative sensor, where information from a satellite system is used for positioning. Another method is to compare information from a *distance measuring equipment* (DME), for instance a range measuring radar sensor, with known environment information from



(a) Surface navigation. (b) UW navigation. (c) Tool positioning.

Figure 1.3: *Positioning: (a) Surface navigation using a radar sensor and a sea chart database, (b) underwater (UW) navigation using a sonar sensor and a depth database, and (c) tool positioning for an industrial robot using accelerometer information at the end-effector.*

databases, i.e., terrain information, sea charts etc. The advantage of this method compared to GPS is that it is insensitive to jamming. In [1, 13, 38, 98] several solutions for a Bayesian terrain aided aircraft navigation problem are proposed. In Figure 1.3, three different positioning applications from the thesis are illustrated. In Figure 1.3 (a) a ship is positioned by comparing information from a sea chart with measured distance from a radar. With a rough model of the dynamics of the ship, the position is found using an estimation algorithm. In Figure 1.3 (b), a similar positioning method is applied to *underwater* (UW) positioning. Instead of a radar sensor, an active acoustic sonar sensor is used. The measured distance to the sea floor is compared to the depth from a depth database. In Figure 1.3 (c), a positioning system for an industrial robot is presented using accelerometer information. Traditionally, industrial robot positioning is based on an accurate model together with measurements from motor angles. Using external information from an accelerometer mounted at the end-effector, positioning can be improved. These applications are presented in Paper C and Paper D. A framework with many positioning applications and models, is given in [38].

1.1.3 Target Tracking

In target tracking the aim is to estimate the kinematic state of an observed object. Target tracking has been an active research area for many years. There exist several good books; for instance [9, 10, 14, 15, 89], where sensor models, target models, and estimation theory are thoroughly described. Tracking models and sensors are also described thoroughly in [75, 76], where the most common types are presented.

Common sensors are radar and IR sensors. If the relative range to the target is measured, it is easier to estimate the target position. If a passive sensor such as the IR-sensor is used, no explicit range information is measured. Hence, it may be

Figure 1.4: *Bearings-only target tracking: Air-to-Air, Air-to-Sea and Sea-to-Sea.*

troublesome to compute the position of the target. By maneuvering the observer, it may be possible to gain observability in both range and range rate. This is often referred to as passive ranging or bearings-only tracking, [15, 42, 89]. In Figure 1.4, passive ranging for Air-to-Air and Air-to-Sea applications using an IR-sensor is illustrated. A Sea-to-Sea bearings-only application for a torpedo is also depicted in Figure 1.4, where a passive acoustic sonar sensor is used. How to incorporation hard constraints using terrain information is discussed in for instance [49]. In [81], littoral tracking, i.e., tracking of targets on land and in sea near the boundary region between them is discussed for a joint tracking and classification problem. These bearings-only applications are discussed in Paper E.

For target tracking applications, modern systems are capable of handling multiple targets. Hence, there will be several measurements available so a data association problem arises, i.e., which measurement to associate with a certain estimated target state. As a result, techniques based on traditional estimation methods have been developed, [9, 15, 87]. When simulation based methods are used, new association methods must be introduced, see for instance [43], or the problem must be formulated in such a manner that classical methods can be used. In Paper F, a tracking application is discussed for multiple targets using the particle filter.

Since the dynamics of the tracked object is not fully known, for instance its input signal (maneuver) is completely unknown, this will affect which models and noise description to use. A particular type of particle filter is discussed in a maneuvering target application using hard-constraints on system states in Paper G.

In active safety systems for cars, estimation of position and velocity for other vehicles is important. In Figure 1.5, a radar-based tracking system is presented. With an accurate estimate of obstacles and other vehicles, collision avoidance or collision mitigation algorithms can be improved. This is discussed in Paper H,

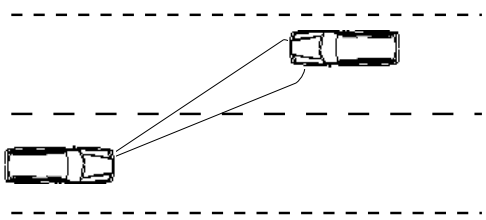


Figure 1.5: *A radar-based automotive tracking system for collision mitigation.*

where a late braking system based on a hypothesis test using Bayesian estimation is discussed.

1.2 Problem Formulation

The purpose of this thesis is to focus on recursive Bayesian estimation using the particle filter, for positioning and target tracking applications. Related problems such as multiple target tracking and sensor fusion are also considered. Positioning systems based on map information together with data from tracking sensors are used to construct positioning systems. Problems with quantized data, real-time performance, and complexity analysis of different methods are fundamental topics. The estimation performance is mainly discussed in terms of the Cramér-Rao lower bound, which is computed analytically or numerically and compared to the actual performance from simulations.

1.3 Outline

Part I contains an overview of estimation theory related to the applications and the theory presented in the thesis. Part of the material has previously been published in [49]. Part II consists of a collection of papers.

1.3.1 Outline of Part I

The objective is to give a short introduction to topics covered by the thesis.

In Chapter 2, common methods in estimation theory are described. Mainly, the maximum likelihood method and the Bayesian method are discussed. In particular, the Bayesian approach for nonlinear dynamic systems is discussed. Several sub-optimal approaches based on Kalman filter theory and multiple models are presented. Fundamental performance limits are discussed in terms of the CRLB.

In Chapter 3, several numerical methods for the estimation and filtering problem are described. Grid-based and stochastic methods for numerical integration are

discussed. Different solutions to recursive estimation problems are presented as variations to the particle filter method.

In Chapter 4, a summary of the thesis is presented.

1.3.2 Outline of Part II

Part II of the thesis consists of the following publications:

Paper A: Filtering and Estimation for Quantized Sensor Information
Classical quantization theory is revisited with the main focus on the measurement quantization. In particular, band-limited noise, moment-based estimation, and dithering noise are discussed. Theoretical performance limits using the Cramér-Rao lower bound are computed. The recursive filtering solution is given by adopting the particle filter for quantized sensor information.

Edited version of the paper:

> R. Karlsson and F. Gustafsson. Filtering and estimation for quantized sensor information. *Submitted to IEEE Transactions on Signal Processing.*

Parts of the paper in:

> R. Karlsson and F. Gustafsson. Particle filtering for quantized sensor information. In *Proceedings of the 13th European Signal Processing Conference*, Antalya, Turkey, September 2005. Submitted as invited paper.

Preliminary version published as Technical Report LiTH-ISY-R-2674, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Paper B: Complexity Analysis of the Marginalized Particle Filter
The computational complexity of the marginalized particle filter is analyzed by counting different operations and estimate their influence on the calculation time.

Edited version of the paper:

> R. Karlsson, T. Schön, and F. Gustafsson. Complexity analysis of the marginalized particle filter. *To appear in IEEE Transactions on Signal Processing.*

Preliminary version published as Technical Report LiTH-ISY-R-2611, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Paper C: Surface and Underwater Navigation using Particle Filters
A framework for two positioning applications at sea is presented. Surface navigation based on range measurements from a radar sensor together with information from a sea chart and an underwater positioning system using sonar measurements and a depth database. The positioning is performed using the particle filter and fundamental limits are derived analytically in terms of the Cramér-Rao lower bound.

Edited version of the paper:

> R. Karlsson and F. Gustafsson. Surface and underwater navigation using particle filters. *Submitted to IEEE Transactions on Signal Processing.*

Parts of the paper in:

R. Karlsson, F. Gustafsson, and T. Karlsson. Particle filtering and Cramér-Rao lower bound for underwater navigation. In *Proceedings IEEE Conference on Acoustics, Speech and Signal Processing*, Hong Kong, April 2003.

R. Karlsson and F. Gustafsson. Particle filter for underwater terrain navigation. In *IEEE Statistical Signal Processing Workshop*, pages 526–529, St. Louis, MO, USA, October 2003. Invited paper.

R. Karlsson and F. Gustafsson. Using the particle filter as mitigation to GPS vulnerability for navigation at sea. In *Proceedings of IEEE Statistical Signal Processing Workshop*, Bordeaux, France, July 2005. Submitted as invited paper.

R. Karlsson and F. Gustafsson. A system and method for surface navigation using radar and sea map. Swedish patent application. Patent pending SE-0400264-8, February 2004.

Preliminary version published as Technical Report LiTH-ISY-R-2649, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Paper D: Bayesian State Estimation of a Flexible Industrial Robot
Positioning of an industrial robot using motor angle measurements together with accelerometer information at the end-effector is discussed. The EKF and the particle filter are implemented, and performance is compared to the Cramér-Rao lower bound numerically. Sensitivity analysis and modeling issues are discussed.

Edited version of the paper:

R. Karlsson and M. Norrlöf. Bayesian state estimation of a flexible industrial robot. *Submitted to IEEE Transactions on Control Systems Technology.*

Parts of the paper in:

R. Karlsson and M. Norrlöf. Bayesian position estimation of an industrial robot using multiple sensors. In *Proceedings of the IEEE Conference on Control Applications*, Taipei, Taiwan, September 2004.

R. Karlsson and M. Norrlöf. Position estimation and modeling of a flexible industrial robot. In *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, July 2005. To appear.

Preliminary version published as Technical Report LiTH-ISY-R-2677, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Paper E: Recursive Bayesian Estimation – Bearings-Only Applications
The bearings-only estimation is addressed in several applications involving Air-to-Air, Air-to-Sea, and Sea-to-Sea scenarios. Simulations and experimental data are used. The comparison involves the particle filter, the marginalized particle filter, and the RPEKF filter bank. Issues such as terrain induced constraints and initialization are discussed.

Edited version of the paper:

R. Karlsson and F. Gustafsson. Recursive Bayesian estimation – bearings-only applications. *Accepted for publication in IEE Proceedings on Radar,*

Sonar, and Navigation. Special issue on target tracking: Algorithms and Applications.

Parts of the paper in:

> R. Karlsson and F. Gustafsson. Range estimation using angle-only target tracking with particle filters. In *Proceedings of American Control Conference*, volume 5, pages 3743–3748, Arlington, Virginia, USA, June 2001. Invited paper.
>
> R. Karlsson. *Simulation Based Methods for Target Tracking*. Linköping Studies in Science and Technology. Licentiate Thesis No. 930, Linköping University, Linköping, Sweden, February 2002.

Paper F: Monte Carlo Data Association for Multiple Target Tracking

The data association problem that arises for a multiple target tracking problem is discussed. When particle filter methods are used instead of extended Kalman filter, new association methods must be developed. A JPDA algorithm is proposed for the particle filter. For many practical applications, the number of particles must be adjusted to reduce computational complexity. This is discussed in terms of a controller for the number of particles.

Edited version of the paper:

> R. Karlsson and F. Gustafsson. Monte Carlo data association for multiple target tracking. In *IEE International Seminar on Target Tracking: Algorithms and Applications*, pages 13/1–13/5, Enschede, The Netherlands, October 2001.

Paper G: Auxiliary Particle Filters for Tracking a Maneuvering Target

The auxiliary particle filter is adopted for the maneuvering target tracking case, with hard constraints on system states. In a simulation study it is compared to the IMM filter.

Edited version of the paper:

> R. Karlsson and N. Bergman. Auxiliary particle filters for tracking a maneuvering target. In *Proceedings of the 39:th IEEE Conference on Decision and Control*, pages 3891–3895, Sydney, Australia, December 2000.

Paper H: Model-based Statistical Tracking and Decision Making for Collision Avoidance Application

Collision mitigation by braking is based on accurate estimates of other vehicles or obstacles. Experimental data is used to model the measurement relation and brake system, and different assumptions are compared in a simulation study using the particle filter and the EKF. The braking criterion is discussed in terms of hypothesis testing for these methods.

Edited version of the paper:

> R. Karlsson, J. Jansson, and F. Gustafsson. Model-based statistical tracking and decision making for collision avoidance application. In *Proceedings of American Control Conference*, Boston, MA, USA, June 2004.

Preliminary version published as Technical Report LiTH-ISY-R-2599, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Papers not Included:

The following paper has contributions from many authors and is therefore not included in the thesis.

Particle Filters for Positioning, Navigation and Tracking

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

1.4 Contributions

The main contributions in the thesis, in order of appearance in Part II, are:

- Analysis of quantization effects in the measurement relation and some quantization theory results. The adoption of the particle filter to measurement quantization.
- Computational complexity analysis of the marginalized particle filter.
- A framework for positioning at sea using information from a distance measuring equipment, featuring an underwater navigation system, and a novel surface navigation system, with analytic Cramér-Rao lower bound calculations.
- A model-based Bayesian approach for position estimation for a flexible industrial robot including modeling, performance analysis, model simplification, and sensitivity analysis.
- A framework for particle filter-based passive ranging incorporating terrain induced constraints in the estimation.
- Data association using the *joint probabilistic data association* (JPDA) algorithm for the particle filter is proposed. A controller for the number of particles is introduced to reduce computational complexity.
- Extension and implementation of a multiple model auxiliary particle filter, using hard constraints for an air traffic control application.
- A collision avoidance algorithm based on a particle filter hypothesis test as well as stochastic integration for an EKF is proposed and a real-time implementation is presented.

2

Estimation – Theory and Methods

The following discrete-time state space description represents a very general dynamic system

$$x_{t+1} = f(x_t, w_t), \tag{2.1a}$$
$$y_t = h(x_t, e_t), \tag{2.1b}$$

where the state vector $x_t \in \mathbb{R}^n$ represents the unknown states or parameters at discrete-time index t, assuming a sample time T. The observation, $y_t \in \mathbb{R}^m$, is often a nonlinear mapping of the current state. Inaccuracies in the system model and in the measurement relation are described by the stochastic processes w_t and e_t. The system states can also be constrained to a subset of the state-space. The main objective in this chapter is to estimate the state, x_t, using ordered observations up to and including time t, $\mathbb{Y}_t = \{y_1, y_2, \ldots, y_t\} = \{y_i\}_{i=1}^t$.

In Section 2.1, two different estimation paradigms are discussed: the Fisherian *maximum likelihood* (ML) method and the Bayesian *maximum a posteriori* (MAP) method. The Bayesian approach is discussed in Section 2.2. In Section 2.3, Kalman filter theory is presented. Estimation techniques related to multiple models are given in Section 2.4. In Section 2.5, fundamental performance limits using the *Cramér-Rao lower bound* (CRLB) are discussed.

2.1 Estimation Paradigms

Estimation techniques are often categorized as Fisherian or Bayesian, [90]. Here the Fisherian *maximum likelihood* (ML) method and the Bayesian *maximum a posteriori* (MAP) method are briefly presented. The ML method was first introduced in [31, 32]. In [2], a detailed historical background is given. The Bayesian paradigm, is based on Bayes' theorem, [11, 44, 90]. In the Bayesian framework, everything unknown is considered as a random variable, whereas for the Fisherian approach, the parameters are considered fixed but unknown. In special cases, the ML and the MAP estimator produce the same estimates, even if the approaches are conceptually different.

Bayes' theorem for a *probability density function* (pdf) with two stochastic variables x and y gives

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}, \tag{2.2}$$

where $p(x|y)$ is the pdf for x given y. Similar, for a set of observations $\mathbb{Y}_t$,

$$p(x|\mathbb{Y}_t) = \frac{p(\mathbb{Y}_t|x)p(x)}{p(\mathbb{Y}_t)} \propto p(\mathbb{Y}_t|x)p(x).$$

For random parameters the point-estimate is found by minimizing the expected value given the observations $\mathbb{Y}_t$

$$\hat{x}_t^{MMS} = \arg\min_x \mathbb{E}\left((\hat{x} - x)^2|\mathbb{Y}_t\right), \tag{2.3}$$

which is denoted the *minimum mean square* (MMS) estimate. It can be shown that the solution is given by the conditional mean

$$\hat{x}_t^{MMS} = \mathbb{E}\left(x|\mathbb{Y}_t\right) = \int x\, p(x|\mathbb{Y}_t)\, dx. \tag{2.4}$$

In [9], this is shown if the conditional expected value is assumed differentiable, since

$$\frac{\partial}{\partial \hat{x}}\mathbb{E}\left((\hat{x} - x)^2|\mathbb{Y}_t\right) = 2(\hat{x} - \mathbb{E}\left(x|\mathbb{Y}_t\right)) = 0 \quad \Rightarrow \quad \hat{x} = \mathbb{E}\left(x|\mathbb{Y}_t\right).$$

The ML method is a statistical method, where a likelihood function is constructed, and the estimate is chosen to maximize a likelihood criterion. If the state vector or parameter vector is given by $x \in \mathbb{R}^n$, the construction of the likelihood function is based on the observations up to present time, $\mathbb{Y}_t$. The likelihood function is constructed by combining likelihoods for different times assuming independence

$$\ell(x|\mathbb{Y}_t) = \prod_{i=1}^{t} p(y_i|x). \tag{2.5}$$

The point estimate of the parameter vector, or state vector, is given by the argument that maximizes the likelihood function. The theory relies on the fact that, asymptotically, the ML estimate converges almost surely to the true value, under fairly general conditions, [74]. In [66], the following definitions are used for the MAP and the ML method

$$\hat{x}^{MAP} = \arg\max_x p(x|\mathbb{Y}_t), \tag{2.6}$$

$$\hat{x}^{ML} = \arg\max_x p(\mathbb{Y}_t|x). \tag{2.7}$$

Example 2.1 (MAP and ML estimation) *Consider independent noisy measurements of a parameter x.*

$$y_i = x + e_i, \quad e_i \sim \mathcal{N}(0, \sigma^2),\ i = 1, 2, \ldots, t.$$

- *The MAP method: If the prior $p(x)$ is assumed Gaussian, $\mathcal{N}(x_0, \sigma_0^2)$, then*

$$p(x|\mathbb{Y}_t) \propto p(\mathbb{Y}_t|x)p(x) = \prod_{i=1}^{t} e^{-\frac{(y_i-x)^2}{2\sigma^2}} e^{-\frac{(x-x_0)^2}{2\sigma_0^2}}.$$

  Maximization yields the MAP estimate as

$$\hat{x}^{MAP} = \arg\max_x p(x|\mathbb{Y}_t) = \frac{\sigma^2 x_0 + \sigma_0^2 \sum_{i=1}^{t} y_i}{t\sigma_0^2 + \sigma^2}.$$

- *The ML method: The likelihood is given by*

$$p(y_i|x) = p_e(y_i - x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_i-x)^2}{2\sigma^2}}.$$

  The ML estimate is given by maximizing the expression

$$\hat{x}^{ML} = \arg\max_x \prod_{i=1}^{t} p(y_i|x) = \frac{1}{t}\sum_{i=1}^{t} y_i.$$

  Note that for $\sigma_0 \to \infty$, the ML and MAP estimates coincide.

Another interesting aspect of MMS and MAP estimation is for the Gaussian case, where it can be shown that $\hat{x}^{MMS} = \hat{x}^{MAP}$. A common technique for state estimation is the *least squares* (LS) method. The idea is to minimize the *mean square error* (MSE), which for a nonlinear function can be formulated as

$$\hat{x}^{LS} = \arg\min_x \sum_{i=1}^{t} \left(y_i - h(x)\right)^2. \tag{2.8}$$

For an overview of different LS-methods and recursive LS-methods, see [47].

Another important issue is when the likelihood function can be decomposed as

$$\ell(x, \mathbb{Y}) = p(\mathbb{Y}|x) = p_1(g(\mathbb{Y}), x)p_2(\mathbb{Y}). \tag{2.9}$$

Hence, it is then clear that the ML estimate depends only on the function $g(\mathbb{Y})$ and not the complete data set $\mathbb{Y}$. The function $g(\mathbb{Y})$ is referred to as the *sufficient statistic* for x, [66]. For a Gaussian pdf, the sufficient statistic is given by the mean and covariance.

2.2 Bayesian Estimation

The problem of estimating a parameter or the state of a nonlinear stochastic system using noisy measurements as observations has been an active research area for many years. In [44], nonlinear and linear problems are discussed. For the special case, with a linear system with additive Gaussian noise, the Kalman filter, [4, 47, 48], yields the finite dimensional recursive solution.

In the Bayesian theory, everything unknown is considered as a stochastic variable. This leads to a description, where the initial or prior distribution is assumed to be known. Using the observations, the estimate can then later be revised by computing the posterior density. The general theory for nonlinear filtering with possible non-Gaussian noise distribution is described thoroughly in [3, 44, 96].

Many discrete-time recursive estimation problems can be formulated using the system presented in (2.1). Given the observations up to time t, $\mathbb{Y}_t$, an estimate, $\hat{x}_{t|t} \in \mathbb{R}^n$, can be calculated from

$$p(x_{t+1}|\mathbb{Y}_t) = \int_{\mathbb{R}^n} p(x_{t+1}|x_t)p(x_t|\mathbb{Y}_t)\, dx_t, \tag{2.10a}$$

$$p(x_t|\mathbb{Y}_t) = \frac{p(y_t|x_t)p(x_t|\mathbb{Y}_{t-1})}{p(y_t|\mathbb{Y}_{t-1})}, \tag{2.10b}$$

for instance as the MMS-estimate. The Bayesian estimation formulation consists of the *time update* (2.10a) and the *measurement update* (2.10b) equations. These equations can be derived using the Markov property, Bayes' theorem and standard calculations from probability theory. The measurement update comes from

$$\begin{aligned} p(x_t|\mathbb{Y}_t) &= \frac{p(\mathbb{Y}_t|x_t)p(x_t)}{p(\mathbb{Y}_t)} = \frac{p(y_t, \mathbb{Y}_{t-1}|x_t)p(x_t)}{p(y_t, \mathbb{Y}_{t-1})} = \frac{p(y_t|x_t, \mathbb{Y}_{t-1})p(\mathbb{Y}_{t-1}|x_t)p(x_t)}{p(y_t|\mathbb{Y}_{t-1})p(\mathbb{Y}_{t-1})} \\ &= \frac{p(y_t|x_t)p(\mathbb{Y}_{t-1}|x_t)p(x_t)}{p(y_t|\mathbb{Y}_{t-1})p(\mathbb{Y}_{t-1})} = \frac{p(y_t|x_t)p(x_t|\mathbb{Y}_{t-1})}{p(y_t|\mathbb{Y}_{t-1})}. \end{aligned} \tag{2.11}$$

In the first equality, Bayes' theorem is used and in the second Bayes' theorem in combination with the definition $\mathbb{Y}_t = \{\mathbb{Y}_{t-1}, y_t\}$. Finally, the Markov property, i.e., $p(y_t|x_t, \mathbb{Y}_{t-1}) = p(y_t|x_t)$, and Bayes' theorem give the result.

The time update equation is given from the following calculations

$$p(x_{t+1}, x_t|\mathbb{Y}_t) = p(x_{t+1}|x_t, \mathbb{Y}_t)p(x_t|\mathbb{Y}_t) = [\text{Markov}] = p(x_{t+1}|x_t)p(x_t|\mathbb{Y}_t). \tag{2.12}$$

Integration of both sides with respect to the state x_t yields,

$$p(x_{t+1}|\mathbb{Y}_t) = \int_{\mathbb{R}^n} p(x_{t+1}|x_t)p(x_t|\mathbb{Y}_t)\, dx_t. \tag{2.13}$$

Often a simplified model with additive noise is assumed

$$x_{t+1} = f(x_t) + w_t, \tag{2.14a}$$
$$y_t = h(x_t) + e_t, \tag{2.14b}$$

where the following relations can be calculated

$$p(x_{t+1}|x_t) = p_{w_t}(x_{t+1} - f(x_t)), \tag{2.15a}$$
$$p(y_t|x_t) = p_{e_t}(y_t - h(x_t)). \tag{2.15b}$$

For the case when additive noise is not assumed, the pdf can be computed locally as shown in the sequel. The analysis and implementation of nonlinear filters are heavily based on expressions involving gradients of scalar functions or vector valued functions. The gradient and Jacobian matrix are defined as:

$$\nabla_x g(x) = \begin{pmatrix} \frac{\partial g}{\partial x_1} \\ \vdots \\ \frac{\partial g}{\partial x_n} \end{pmatrix}, \; g : \mathbb{R}^n \mapsto \mathbb{R}, \tag{2.16a}$$

$$\nabla_x g^T(x) = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_m}{\partial x_1} \\ \vdots & & \vdots \\ \frac{\partial g_1}{\partial x_n} & \cdots & \frac{\partial g_m}{\partial x_n} \end{pmatrix}, \; g : \mathbb{R}^n \mapsto \mathbb{R}^m. \tag{2.16b}$$

Also, the Laplacian for the scalar function $g(x, y)$ with $x \in \mathbb{R}^n, y \in \mathbb{R}^m$ is defined as

$$\boldsymbol{\Delta}_y^x g(x, y) = \nabla_y (\nabla_x g(x, y))^T, \; g : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}. \tag{2.17}$$

For the case when additive noise is not assumed, the following theorem from [44] is applicable.

Theorem 2.1 *Let X and Y be two random vectors, with $Y = g(X)$. Suppose that the inverse g^{-1} exists, and both g and g^{-1} are continuously differentiable. Then*

$$p_Y(y) = p_X(g^{-1}(y)) \cdot \left|\det\left(\nabla_y (g^{-1}(y))^T\right)\right|, \tag{2.18}$$

where $\left|\det\left(\nabla_y (g^{-1}(y))^T\right)\right|$ denotes the absolute value of the Jacobian determinant.

Proof See [44]. □

By applying Theorem 2.1, as suggested in [44], the transition pdf and likelihood pdf for the more general case in (2.1) are given as

$$p(x_{t+1}|x_t) = p_{w_{t+1}}\left(f^{-1}(x_t, x_{t+1})\right) \cdot \left|\det\left(\nabla_{x_{t+1}}(f^{-1})^T\right)\right|, \tag{2.19a}$$

$$p(y_t|x_t) = p_{e_t}\left(h^{-1}(y_t, x_t)\right) \cdot \left|\det\left(\nabla_{y_t}(h^{-1})^T\right)\right|, \tag{2.19b}$$

using the invertability

$$w_t = f^{-1}(x_t, x_{t+1}), \tag{2.20a}$$

$$e_t = h^{-1}(y_t, x_t). \tag{2.20b}$$

2.3 Kalman Filters

In general, there does not exist a finite dimensional solution to the Bayesian estimation problem in (2.10). However, for the case of a linear system subject to Gaussian noise the solution can be formulated recursively using the *Kalman filter* (KF). This is discussed in Section 2.3.1. For nonlinear problems, the system can be linearized before applying the Kalman filter. The linearization is done around the estimated state. This is referred to as the *extended Kalman filter* (EKF) and discussed in Section 2.3.2. Instead of first linearizing the system in order to apply the Kalman filter, the estimation problem can be approximated directly. The pdfs given in (2.10) are then solved numerically. This can be done using deterministic or stochastic methods. The solution using stochastic integration leads to the *particle filter* formulation, which will be discussed in detail in Chapter 3.

2.3.1 The Kalman Filter

If a linear system with additive Gaussian noise is assumed, there exists a finite dimensional solution to the Bayesian time update and measurement update equations (2.10), given by the *Kalman filter* (KF), [47, 48]. Since the system is linear and Gaussian, the update formula will remain Gaussian, and since a Gaussian distribution can be described by its first two moments (mean and covariance), the update equations will consist of mean and covariance updates only. Many books describing different aspects of the Kalman filter exist. Much of the classical theory is described in [4, 47]. Consider a linear-Gaussian time-varying state space model:

$$x_{t+1} = F_t x_t + G_{u,t} u_t + G_{w,t} w_t, \tag{2.21a}$$

$$y_t = H_t x_t + e_t. \tag{2.21b}$$

The input signal is denoted u_t, the process noise w_t and the measurement noise e_t, with $\mathrm{Cov}\,(w_t) = Q_t$ and $\mathrm{Cov}\,(e_t) = R_t$ respectively, and with cross covariance S_t. This is summarized as:

$$\mathbb{E}\left(\begin{pmatrix} w_t \\ e_t \\ x_0 \end{pmatrix}\begin{pmatrix} w_s^T & e_s^T & x_0^T & 1 \end{pmatrix}\right) = \begin{pmatrix} Q_{t,s}\delta_{t,s} & S_{t,s}\delta_{t,s} & 0 & 0 \\ S_{t,s}^T\delta_{t,s} & R_{t,s}\delta_{t,s} & 0 & 0 \\ 0 & 0 & \Pi_0 & 0 \end{pmatrix}, \tag{2.22}$$

where

$$\delta_{t,s} = \begin{cases} 1, & t = s, \\ 0, & t \neq s. \end{cases} \tag{2.23}$$

If the initial state x_0 and process noise w_t and measurement noise e_t are Gaussian variables, then given the cumulative set of observations, $\mathbb{Y}_t$,

$$x_{t+1}|\mathbb{Y}_t \sim \mathcal{N}(\hat{x}_{t+1|t}, P_{t+1|t}), \tag{2.24a}$$
$$x_t|\mathbb{Y}_t \sim \mathcal{N}(\hat{x}_{t|t}, P_{t|t}), \tag{2.24b}$$

Consider first the case of un-correlated noises, $S_t = 0$. The time and measurement updates in the Kalman filter are given in Algorithm 2.1. The case of correlated noise, i.e., $S_t \neq 0$, can be handled using a Gram-Schmidt orthogonalization process, [47, p.326].

Algorithm 2.1 Kalman filter (KF)

Assume that $\hat{x}_{0|-1} = x_0$ and $P_{0|-1} = \Pi_0$.
Time update:

$$\hat{x}_{t+1|t} = F_t\hat{x}_{t|t} + G_{u,t}u_t, \tag{2.25a}$$
$$P_{t+1|t} = F_tP_{t|t}F_t^T + G_{w,t}Q_tG_{w,t}^T, \tag{2.25b}$$

Measurement update:

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - H_t\hat{x}_{t|t-1}), \tag{2.26a}$$
$$P_{t|t} = P_{t|t-1} - K_tH_tP_{t|t-1}, \tag{2.26b}$$

where

$$K_t = P_{t|t-1}H_t^T\left(H_tP_{t|t-1}H_t^T + R_t\right)^{-1}. \tag{2.27}$$

2.3.2 The Extended Kalman Filter

Many estimation problems are nonlinear, but the noise model is assumed Gaussian. The main idea is to linearize the system and apply the Kalman filter. This approach is referred to as the *extended Kalman filter* (EKF), [4, 47], or Schmidt extended Kalman filter.

Consider the following system

$$x_{t+1} = f(x_t) + g(x_t)w_t, \tag{2.28a}$$
$$y_t = h(x_t) + e_t, \tag{2.28b}$$

where w_t and e_t are considered Gaussian with zero mean and covariances Q_t and R_t respectively, and initial uncertainty Π_0, i.e.,

$$\mathbb{E}\left(\begin{pmatrix} w_t \\ e_t \\ x_0 - \bar{x}_0 \end{pmatrix} \begin{pmatrix} w_s^T & e_s^T & (x_0 - \bar{x}_0)^T & 1 \end{pmatrix}\right) = \begin{pmatrix} Q_{t,s}\delta_{t,s} & 0 & 0 & 0 \\ 0 & R_{t,s}\delta_{t,s} & 0 & 0 \\ 0 & 0 & \Pi_0 & 0 \end{pmatrix}. \tag{2.29}$$

The system can also be time-varying, but that is not explicitly stated here for notational simplicity. Use the following approximations

$$f(x_t) \approx f(\hat{x}_{t|t}) + F_t(x_t - \hat{x}_{t|t}), \tag{2.30a}$$
$$h(x_t) \approx h(\hat{x}_{t|t-1}) + H_t(x_t - \hat{x}_{t|t-1}), \tag{2.30b}$$
$$g(x_t) \approx g(\hat{x}_{t|t}) = G_t, \tag{2.30c}$$

where

$$F_t^T = \nabla_x f^T(x)\big|_{x=\hat{x}_{t|t}}, \quad H_t^T = \nabla_x h^T(x)\big|_{x=\hat{x}_{t|t-1}}. \tag{2.31}$$

Combining the system and the approximations now gives

$$x_{t+1} = F_t x_t + \underbrace{\big(f(\hat{x}_{t|t}) - F_t\hat{x}_{t|t}\big)}_{\text{known at time } t} + G_t w_t, \tag{2.32a}$$
$$y_t - \underbrace{\big(h(\hat{x}_{t|t-1}) - H_t\hat{x}_{t|t-1}\big)}_{\text{known at time } t-1} = H_t x_t + e_t. \tag{2.32b}$$

This is a linear state-space model for x_t. Applying the Kalman filter yields

$$\hat{x}_{t+1|t} = F_t\hat{x}_{t|t} + \big(f(\hat{x}_{t|t}) - F_t\hat{x}_{t|t}\big) = f(\hat{x}_{t|t}), \tag{2.33a}$$
$$\begin{aligned}\hat{x}_{t|t} &= \hat{x}_{t|t-1} + K_t\big(y_t - (h(\hat{x}_{t|t-1}) - H_t\hat{x}_{t|t-1}) - H_t\hat{x}_{t|t-1}\big) \\ &= \hat{x}_{t|t-1} + K_t\big(y_t - h(\hat{x}_{t|t-1})\big).\end{aligned} \tag{2.33b}$$

The covariance recursion and Kalman gain are given by the Kalman filter. To summarize, the EKF is given in Algorithm 2.2.

2.4 Multiple Models

Many estimation problems involve rapid changes in the system dynamics. In for instance target tracking applications, this can be due to the unknown target maneuver sequence. To achieve an accurate estimate, several models can be used, each adopted to describe a specific feature. One important issue for a multiple model or filter application, is to reduce the number of hypotheses. This can be done using pruning or merging (mixing) techniques. The estimate is constructed either by mixing or switching between the models, [10, 37].

Algorithm 2.2 Extended Kalman filter (EKF)

Time update :

$$\hat{x}_{t+1|t} = f(\hat{x}_{t|t}), \tag{2.34a}$$
$$P_{t+1|t} = F_t P_{t|t} F_t^T + G_t Q_t G_t^T, \tag{2.34b}$$

Measurement update :

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t \left(y_t - h(\hat{x}_{t|t-1}) \right), \tag{2.35a}$$
$$P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}, \tag{2.35b}$$

where

$$K_t = P_{t|t-1} H_t^T \left(H_t P_{t|t-1} H_t^T + R_t \right)^{-1}, \tag{2.36a}$$
$$F_t^T = \nabla_x f^T(x)\big|_{x=\hat{x}_{t|t}}, \quad H_t^T = \nabla_x h^T(x)\big|_{x=\hat{x}_{t|t-1}}. \tag{2.36b}$$

Briefly summarizing the multiple model approach using merging techniques, assume N different models/filters approximating the density $p(x)$ by mean $\hat{x}^{(i)}$ and covariance $P^{(i)}$. The merging is then performed according to

$$p(x) = \sum_{i=1}^{N} \gamma^{(i)} \mathcal{N}(x; \hat{x}^{(i)}, P^{(i)}) \approx \mathcal{N}(x; \hat{x}, P), \quad \sum_{i=1}^{N} \gamma^{(i)} = 1, \tag{2.37}$$

where

$$\hat{x} = \sum_{i=1}^{N} \gamma^{(i)} \hat{x}^{(i)}, \tag{2.38a}$$
$$P = \sum_{i=1}^{N} \gamma^{(i)} \left(P^{(i)} + (\hat{x}^{(i)} - \hat{x})(\hat{x}^{(i)} - \hat{x})^T \right). \tag{2.38b}$$

The term $\sum_{i=1}^{N} \gamma^{(i)} (x^{(i)} - \hat{x})(x^{(i)} - \hat{x})^T$ is referred to as the *spread of the mean.*

In this thesis two different techniques are used. The *interacting multiple model* (IMM) described in Section 2.4.1 and the *range parameterized extended Kalman filter* (RPEKF) described in Section 2.4.2. In Section 2.4.3, several other multiple model methods are briefly discussed.

2.4.1 Interactive Multiple Models

In [16], a filtering algorithm for linear discrete-time filters with Markovian coefficients is given. The suboptimal filter is called the *interacting multiple model* (IMM). The presentation here is based on material from [10, p. 463-464].

Assume that N different models at time t are used, denoted $\mathcal{M}_t^{(i)}$, where the probability for each model is $\gamma_t^{(i)} = \text{Prob}(\mathcal{M}_t^{(i)}|\mathbb{Y}_t)$. The pdf at time t is given by the total probability theorem using N different models as

$$p(x_t|\mathbb{Y}_t) = \sum_{j=1}^{N} p(x_t|\mathcal{M}_t^{(j)}, \mathbb{Y}_t) \underbrace{\text{Prob}(\mathcal{M}_t^{(j)}|\mathbb{Y}_t)}_{\gamma_t^{(j)}}. \tag{2.39}$$

Applying Bayes' theorem to the first factor in (2.39) using $\mathbb{Y}_t = \{y_t, \mathbb{Y}_{t-1}\}$ gives

$$p(x_t|\mathcal{M}_t^{(j)}, \mathbb{Y}_t) \propto p(y_t|\mathcal{M}_t^{(j)}, x_t) p(x_t|\mathcal{M}_t^{(j)}, \mathbb{Y}_{t-1}). \tag{2.40}$$

Applying the total probability theorem to the last factor in (2.40) gives

$$\begin{aligned} p(x_t|\mathcal{M}_t^{(j)}, \mathbb{Y}_{t-1}) &= \sum_{i=1}^{N} p(x_t|\mathcal{M}_t^{(j)}, \mathcal{M}_{t-1}^{(i)}, \mathbb{Y}_{t-1}) \underbrace{\text{Prob}(\mathcal{M}_{t-1}^{(i)}|\mathcal{M}_t^{(j)}, \mathbb{Y}_{t-1})}_{\gamma_{t-1|t-1}(i,j)} \\ &\approx \sum_{i=1}^{N} p\left(x_t|\mathcal{M}_t^{(j)}, \mathcal{M}_{t-1}^{(i)}, \{\hat{x}_{t-1|t-1}^{(l)}, P_{t-1|t-1}^{(l)}\}_{l=1}^{N}\right) \gamma_{t-1|t-1}(i,j) \\ &= \sum_{i=1}^{N} p(x_t|\mathcal{M}_t^{(j)}, \mathcal{M}_{t-1}^{(i)}, \hat{x}_{t-1|t-1}^{(i)}, P_{t-1|t-1}^{(i)}) \gamma_{t-1|t-1}(i,j). \end{aligned} \tag{2.41}$$

The approximation in (2.41) is due to the fact that the models summarize the history through the estimates and covariances. The mixing probabilities are expressed using Bayes' theorem as

$$\begin{aligned} \gamma_{t-1|t-1}(i,j) &= \text{Prob}(\mathcal{M}_{t-1}^{(i)}|\mathcal{M}_t^{(j)}, \mathbb{Y}_{t-1}) \\ &\propto \underbrace{\text{Prob}(\mathcal{M}_t^{(j)}|\mathcal{M}_{t-1}^{(i)}, \mathbb{Y}_{t-1})}_{p(i,j)} \underbrace{\text{Prob}(\mathcal{M}_{t-1}^{(i)}|\mathbb{Y}_{t-1})}_{\gamma_{t-1}^{(i)}}, \end{aligned} \tag{2.42}$$

where $p(i,j)$ in practice is used as a design parameter. Condensing (2.41) by approximating the Gaussian mixture with a single Gaussian gives

$$\begin{aligned} p(x_t|\mathcal{M}_t^{(j)}, \mathbb{Y}_{t-1}) &= \sum_{j=1}^{N} \mathcal{N}\left(x_t; \mathbb{E}\left(x_t|\mathcal{M}_t^{(j)}, \hat{x}_{t-1|t-1}^{(i)}\right), \text{Cov}\left(\cdot\right)\right) \gamma_{t-1|t-1}(i,j) \\ &\approx \mathcal{N}\left(x_t; \sum_{i=1}^{N} \mathbb{E}\left(x_t|\mathcal{M}_t^{(j)}, \hat{x}_{t-1|t-1}^{(i)}\right) \gamma_{t-1|t-1}(i,j), \text{Cov}\left(\cdot\right)\right) \\ &= \mathcal{N}\left(x_t; \mathbb{E}\left(x_t|\mathcal{M}_t^{(j)}, \sum_{i=1}^{N} \hat{x}_{t-1|t-1}^{(i)} \gamma_{t-1|t-1}(i,j)\right), \text{Cov}\left(\cdot\right)\right), \end{aligned}$$

where $\text{Cov}\left(\cdot\right)$ denotes the covariance for each expression.

The IMM method is summarized in Algorithm 2.3, where also all covariance matrices are explicitly given.

Algorithm 2.3 Interactive Multiple Model (IMM), [10]

1: Calculate the mixing probabilities, using the probability that mode $\mathcal{M}^{(i)}$ is in effect at $t-1$ given that model $\mathcal{M}^{(j)}$ is in effect at time t conditioned upon the measurements $\mathbb{Y}_{t-1}$.

$$\begin{aligned}\gamma_{t-1|t-1}(i,j) &= \text{Prob}(\mathcal{M}_{t-1}^{(i)}|\mathcal{M}_t^{(j)},\mathbb{Y}_{t-1}),\\ &= \frac{1}{\bar{c}^{(j)}}\underbrace{\text{Prob}(\mathcal{M}_t^{(j)}|\mathcal{M}_{t-1}^{(i)},\mathbb{Y}_{t-1})}_{p(i,j)}\underbrace{\text{Prob}(\mathcal{M}_{t-1}^{(i)}|\mathbb{Y}_{t-1})}_{\gamma_{t-1}^{(i)}},\\ & i,j=1,\ldots,N,\\ \bar{c}^{(j)} &= \sum_{i=1}^{N} p(i,j)\gamma_{t-1}^{(i)}, \qquad j=1,\ldots,N.\end{aligned}$$

2: Calculate the initial mixing condition for $j=1,\ldots,N$ filters

$$\begin{aligned}\hat{x}_{t-1|t-1}^{(j),0} &= \sum_{i=1}^{N}\hat{x}_{t-1|t-1}^{(i)}\gamma_{t-1|t-1}(i,j),\\ D_{t-1}(i,j) &= \hat{x}_{t-1|t-1}^{(i)} - \hat{x}_{t-1|t-1}^{(j),0},\\ P_{t-1|t-1}^{(j),0} &= \sum_{i=1}^{N}\gamma_{t-1|t-1}(i,j)\left(P_{t-1|t-1}^{(i)} + D_{t-1}(i,j)D_{t-1}^{T}(i,j)\right).\end{aligned}$$

3: Calculate $\hat{x}_{t|t}^{(j)}, P_{t|t}^{(j)}$ for $j=1,\ldots,N$, with likelihood functions

$$\ell_t^{(j)} = p(y_t|\mathcal{M}_t^{(j)},\hat{x}_{t-1|t-1}^{(j),0},P_{t-1|t-1}^{(j),0}).$$

4: Mode probability update for $j=1,\ldots,N$

$$\begin{aligned}\gamma_t^{(j)} &= \text{Prob}(\mathcal{M}_t^{(j)}|\mathbb{Y}_t) = \frac{1}{c}p(y_t|\mathcal{M}_t^{(j)},\mathbb{Y}_{t-1})\,\text{Prob}(\mathcal{M}_t^{(j)}|\mathbb{Y}_{t-1})\\ &= \frac{1}{c}\ell_t^{(j)}\sum_{i=1}^{N}p(i,j)\gamma_{t-1}^{(i)} = \frac{1}{c}\ell_t^{(j)}\bar{c}^{(j)},\\ c &= \sum_{j=1}^{N}\ell_t^{(j)}\bar{c}^{(j)}.\end{aligned}$$

5: Estimate and covariance combination

$$\hat{x}_{t|t} = \sum_{j=1}^{N}\gamma_t^{(j)}\hat{x}_{t|t}^{(j)}, \quad P_{t|t} = \sum_{j=1}^{N}\gamma_t^{(j)}\left(P_{t|t}^{(j)} + (\hat{x}_{t|t}^{(j)} - \hat{x}_{t|t})(\hat{x}_{t|t}^{(j)} - \hat{x}_{t|t})^T\right).$$

In Example 2.2, a radar tracking application is presented using the IMM method with two filters. One filter is used to handle straight paths accurately, whereas the other is used to manage maneuvers. Due to the nonlinearities in the measurement equation an EKF is used for the estimation.

Example 2.2 (The IMM method for two models) *In Figure 2.1, the IMM algorithm is presented graphically for the special case, where only 2 models are used. Consider a radar tracking system, where the distance and angle to the target*

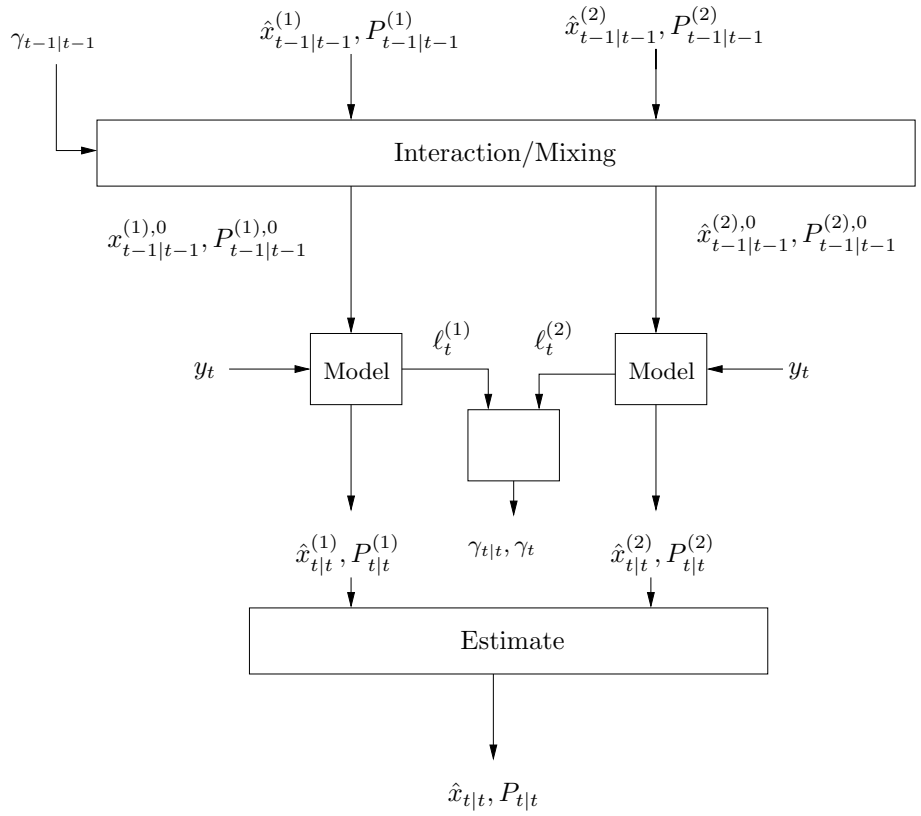


Figure 2.1: *Schematic overview of the IMM algorithm for 2 models.*

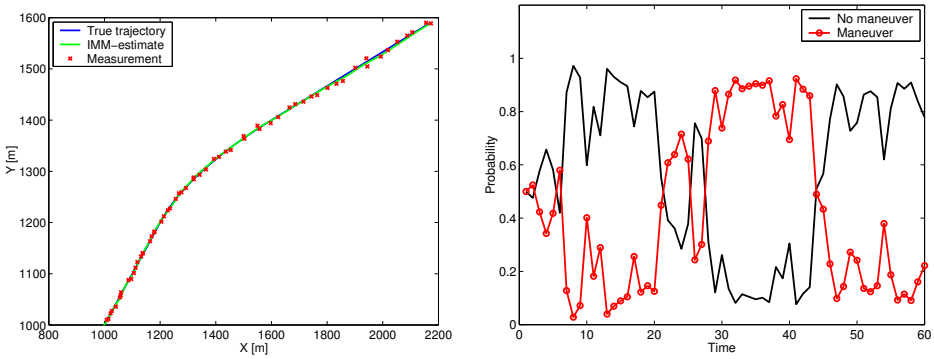
are measured from a ground based system placed in the origin. The following Cartesian model with state vector $x_t = (X\ Y\ \dot{X}\ \dot{Y})^T$*, where* X, Y *are the Cartesian position coordinates and* $\dot{X}, \dot{Y}$ *are the velocity components. The discrete-time*

model and measurement relation are given as

$$x_{t+1} = \begin{pmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} x_t + \begin{pmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{pmatrix} w_t,$$

$$y_t = h(x_t) = \begin{pmatrix} \sqrt{X_t^2 + Y_t^2} \\ \arctan(\frac{Y_t}{X_t}) \end{pmatrix} + e_t, \quad e_t \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 10^2 & 0 \\ 0 & 0.001^2 \end{pmatrix} \right).$$

The true trajectory is generated as a straight path, followed by a maneuver between $t = 20 - 40$ *and then continuing on a straight path. In the IMM filter, the two models are tuned to maneuver and non-maneuver, using different values on the process noise covariance matrix,* $Q_t = \mathrm{Cov}\,(w_t)$*, (low variance for straight path and high variance for maneuver). The probability to change between the models is* 0.05*. In Figure 2.2 (a), the scenario, true trajectory, measurements, and IMM estimates are given and in Figure 2.2 (b), the probabilities from the IMM filter (maneuver or non-maneuver) are presented.*



(a) Scenario and IMM estimates

(b) IMM probabilities

Figure 2.2: *The IMM-2 filter estimates and probabilities.*

In Paper G, the IMM filter is used in a maneuvering target tracking application.

2.4.2 The Range Parameterized Extended Kalman Filters

Passive ranging or estimation of range and velocity using only a passive sensor, such as the IR sensor, is a difficult problem. The main idea is to maneuver the own platform in such a way that relative range and velocity can be estimated. In practice, this means that maneuvers have to be performed to out-maneuver the unknown target. An accurate navigation system is also assumed based on information from reliable sensors with small errors, so the own trajectory and movement can be

accurately estimated. Several approaches exist to estimate the range using a single tracking filter. As described in [92, 95, 97] a modified spherical/polar coordinate system is preferred to a traditional Cartesian system. The approach in this section is to study a multiple model method to estimate the unknown range and velocity. A special method called the *range parameterized extended Kalman filter* (RPEKF) is used, which consists of a bank of EKFs, each tuned to a certain range. The presentation in this section follows the development in [55]. The RPEKF method described in [8, 73] consists of a bank of extended Kalman filters in Cartesian coordinates, initialized to different range assumptions for the angle-only tracking application. In [85], the filter bank is expressed in modified polar coordinates.

For a particular EKF the performance is dependent on the *coefficient of variation*, C_R, [85]. To have comparable performance for each filter, the same C_R value should be used on each interval. Approximatively, this is given as $\sigma^{(i)}/r^{(i)}, i = 1, \ldots, N_F$, where $r^{(i)}$ and $\sigma^{(i)}$ are the range and standard deviation for the different filters. In Figure 2.3, the range intervals are depicted for a predefined interval $(r_{\min}, r_{\max})$. The intervals and the C_R are given as

$$r^{(i)} = \frac{r_{\min}}{2}(\rho^i + \rho^{i-1}), \quad \rho = \left(\frac{r_{\max}}{r_{\min}}\right)^{1/N_F}, \tag{2.43a}$$

$$C_R = \frac{\sigma^{(i)}}{r^{(i)}} = \frac{2(\rho - 1)}{\sqrt{12}(\rho + 1)}, \tag{2.43b}$$

Therefore, the variance for each interval is given as $\sigma^{(i)} = r^{(i)} C_R$. The RPEKF

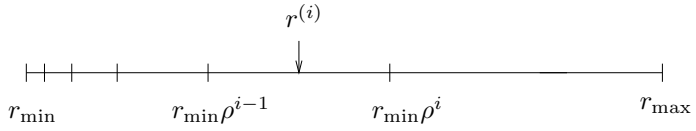


Figure 2.3: *RPEKF range intervals.*

uses the likelihood from each EKF, to recursively update its probability according to

$$\gamma_t^{(i)} = p(y_t|i)\gamma_{t-1}^{(i)}. \tag{2.44}$$

The prior distribution is assumed uniform, i.e., $\gamma_0^{(i)} = 1/N_F$, $i = 1, \ldots, N_F$. However, if other information is available it could be used to enhance the performance. Under a Gaussian assumption, the likelihood is given from the EKF as

$$p(y_t|i) \propto \frac{1}{\sqrt{\det(S_t^{(i)})}} e^{-\frac{1}{2}\left(y_t - h(\hat{x}_{t|t-1}^{(i)})\right)^T \left(S_t^{(i)}\right)^{-1} \left(y_t - h(\hat{x}_{t|t-1}^{(i)})\right)}, \tag{2.45a}$$

$$S_t^{(i)} = H_t^{(i)} P_{t|t-1}^{(i)} (H_t^{(i)})^T + R_t, \tag{2.45b}$$

$$\left(H_t^{(i)}\right)^T = \nabla_x h^T(x)\big|_{x=\hat{x}_{t|t-1}^{(i)}}, \tag{2.45c}$$

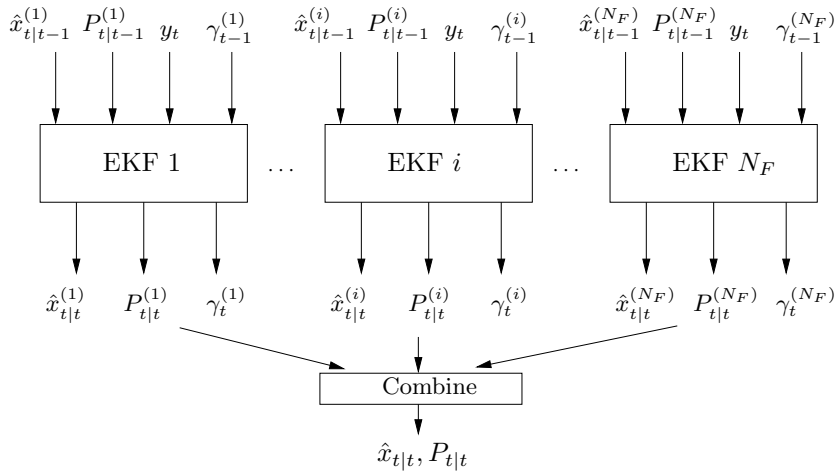


Figure 2.4: *The RPEKF structure.*

where R_t is the measurement noise covariance matrix. The measurement update for each filter is given by the Kalman filter equations. The combined estimate and covariance can now be expressed using (2.38)

$$\hat{x}_{t|t} = \sum_{i=1}^{N_F} \gamma_t^{(i)} \hat{x}_{t|t}^{(i)}, \tag{2.46a}$$

$$P_{t|t} = \sum_{i=1}^{N_F} \gamma_t^{(i)} \left(P_{t|t}^{(i)} + (\hat{x}_{t|t}^{(i)} - \hat{x}_{t|t})(\hat{x}_{t|t}^{(i)} - \hat{x}_{t|t})^T \right), \tag{2.46b}$$

where $P_{t|t}^{(i)}$ is the covariance and $\hat{x}_{t|t}^{(i)}$ the estimate for the different range filters $i = 1, \ldots, N_F$. In Figure 2.4, the RPEKF idea is summarized. If the filter probability is less than a predefined threshold or if some other criterion, such as if the estimated range in a filter is outside the $(r_{\min}, r_{\max})$ interval, the filter is removed from further calculations.

In paper E, the RPEKF is applied to several bearings-only applications.

2.4.3 Other Multiple Model Methods

In this thesis only the previously described IMM and RPEKF methods are used in the applications. As mentioned, other important multiple model methods exist. A few of them are described below.

The *adaptive forgetting through multiple models* (AFMM), [5], is an example of an estimation technique using multiple models/filters. Instead of mixing the individual estimates, it is computed by jumping between the models. By applying

a change detector between different models, the switching can be implemented differently.

The general multiple model idea is often based on the *Gaussian sum* (GS) approximation, described in [4, 96]. The GS method approximates the pdf with a sum of Gaussian densities.

A similar method as the IMM is given by the *generalized pseudo-Bayesian* (GPB) method. The GPB approach merges the mixture after the measurement update, whereas for the IMM method merging is applied after the time update of the weights rather than after the measurement update. In [37], these methods are described in more detail.

The *unscented Kalman filter* (UKF), [45, 46], uses the unscented transform for state estimation. Unlike the EKF the system is not linearized, instead the posterior density is approximated by a Gaussian density. Combination of different approaches, such as IMM-EKF, IMM-UKF etc. exist.

In [37, 89] several of the methods presented in this section are described in detail. In Section 3.2.5 several of these methods are discussed in the particle filter framework.

2.5 Cramér-Rao Lower Bound

It is often important to know the theoretical performance of an estimator or filter given a model structure. An often used performance bound is given by the *Fisher information matrix* (FIM), which gives a bound on the second order moment. This bound is often referred to as the *Cramér-Rao lower bound* (CRLB). In this section, the CRLB is presented for static and dynamic systems.

In Paper A and Paper C theoretical CRLB limits are presented. In most of the other papers the performance in terms of CRLB is evaluated numerically.

2.5.1 Cramér-Rao Lower Bound for a Static System

For an unbiased estimator, $\mathbb{E}(\hat{x}) = x$, the CRLB, [21, 66, 74], is given by

$$\mathrm{Cov}(x - \hat{x}) = \mathbb{E}\left((x - \hat{x})(x - \hat{x})^T\right) \succeq J^{-1}(x), \tag{2.47a}$$

$$J(x) = \mathbb{E}\left(-\boldsymbol{\Delta}_x^x \log p(y|x)\right), \tag{2.47b}$$

where $J(x)$ denotes the FIM for the measurement y regarding the stochastic parameter x. The Laplacian operator is defined in (2.17). An equivalent representation of the information is, [66],

$$\mathbb{E}\left(\nabla_x \log p(y|x)(\nabla_x \log p(y|x))^T\right) = \mathbb{E}\left(-\boldsymbol{\Delta}_x^x \log p(y|x)\right). \tag{2.48}$$

The gradient and Jacobian are defined in (2.16). Particularly, a Gaussian likelihood $p(y|x)$, with measurement covariance R, gives

$$J(x) = H^T(x) R^{-1} H(x), \tag{2.49}$$

where

$$H^T(x) = \nabla_x h^T(x). \tag{2.50}$$

For the case with multiple independent measurements $y^{(i)}, i = 1, \ldots, M$, the information is given as

$$J(x) = \sum_{i=1}^{M} J^{(i)}(x), \tag{2.51}$$

due to the additivity of information, where $J^{(i)}$ is the information for measurement i.

2.5.2 Cramér-Rao Lower Bound for a Dynamic System

In this section, a CRLB bound for the filtering case is presented. The theoretical posterior CRLB for a dynamic system is analyzed in [13, 26, 100, 102]. Here, the following model is considered:

$$x_{t+1} = f(x_t, w_t), \tag{2.52a}$$
$$y_t = h(x_t) + e_t. \tag{2.52b}$$

From [13], the posterior CRLB is

$$\mathrm{Cov}\left(x_t - \hat{x}_{t|t}\right) = \mathbb{E}\left((x_t - \hat{x}_{t|t})(x_t - \hat{x}_{t|t})^T\right) \succeq P_{t|t}, \tag{2.53}$$

where $P_{t|t}$ can be retrieved from the recursion

$$P_{t+1|t+1}^{-1} = Q_t^{-1} + J_{t+1} - S_t^T \left(P_{t|t}^{-1} + V_t\right)^{-1} S_t, \tag{2.54}$$

assuming that Q_t is invertible and where

$$V_t = \mathbb{E}\left(-\boldsymbol{\Delta}_{x_t}^{x_t} \log p(x_{t+1}|x_t)\right), \tag{2.55a}$$
$$S_t = \mathbb{E}\left(-\boldsymbol{\Delta}_{x_t}^{x_{t+1}} \log p(x_{t+1}|x_t)\right), \tag{2.55b}$$
$$Q_t^{-1} = \mathbb{E}\left(-\boldsymbol{\Delta}_{x_{t+1}}^{x_{t+1}} \log p(x_{t+1}|x_t)\right), \tag{2.55c}$$
$$J_t = \mathbb{E}\left(-\boldsymbol{\Delta}_{x_t}^{x_t} \log p(y_t|x_t)\right). \tag{2.55d}$$

For linear dynamics with additive Gaussian noise

$$x_{t+1} = F_t x_t + w_t, \tag{2.56}$$

the following holds

$$V_t = F_t Q_t^{-1} F_t^T, \quad S_t = -F_t^T Q_t^{-1}, \tag{2.57}$$

where $\mathrm{Cov}\,(w_t) = Q_t$. If Q_t is not full rank, instead of using the above information form, the CRLB can be expressed with a Kalman filter recursion for the covariance matrix.

In Paper A and Paper C theoretical CRLB limits are presented. If the model is too complicated the CRLB can be calculated numerically. Also, the square root of the trace of the CRLB matrix can be compared to the *root mean square error* (RMSE),

$$\mathrm{RMSE}(t) = \left(\frac{1}{N_{\mathrm{MC}}} \sum_{j=1}^{N_{\mathrm{MC}}} \| x_t^{\mathrm{TRUE}} - \hat{x}_t^{(j)} \|_2^2 \right)^{1/2}, \tag{2.58}$$

where x_t^{TRUE} is the true state at time t and $\hat{x}_t^{(j)}$ the estimate from the chosen estimator in Monte Carlo simulation j.

3

Numerical Methods for Bayesian Estimation

In this chapter, the main objective is to study methods to solve the untractable recursive Bayesian estimation problem. The solution is based on numerical integration. First a deterministic grid-based integration method is discussed and then a stochastic integration method is introduced. The main part of the chapter is devoted to recursive Bayesian estimation based on the *particle filter* (PF) method.

In Section 3.1 deterministic and stochastic numerical integration are discussed as a motivation for the recursive solution of the Bayesian estimation problem using the particle filter presented in Section 3.2.

3.1 Numerical Integration

Several different approaches exist for solving integrals numerically. Many methods are based on deterministic integration, where the integral is approximated by a Riemann sum, others rely on stochastic simulation. Sometimes the method is only applicable to integration on $\mathbb{R}$, whereas other methods are more general.

3.1.1 Grid-Based Integration

The grid-based approximation for a general integral in $\mathbb{R}^n$ is defined as

$$\int_{\mathbb{R}^n} g(x_t)\,dx_t \approx \sum_{i=1}^{N} g(x_t^{(i)})\Delta^n, \tag{3.1}$$

using a regular grid, where Δ^n represents the volume and where $\{x_t^{(i)}\}_{i=1}^N$ is the set of samples. The approximation error is dependent on the grid size, Δ, hence in principle on the dimension of the state space.

In [72], the Bayesian approach is investigated for a discrete-time nonlinear system. The Bayesian time update and measurement update are solved using an approximative numerical method, where the density functions are piecewise constant on regular regions in the state space. Applying this grid-based integration to the general Bayesian estimation problem in (2.10), using the model with additive noise as in (2.14) yields the following approximation

$$p(x_t^{(i)}|\mathbb{Y}_t) = \gamma_t^{-1} p_{e_t}(y_t - h(x_t^{(i)}))p(x_t^{(i)}|\mathbb{Y}_{t-1}), \tag{3.2a}$$

$$p(x_{t+1}^{(i)}|\mathbb{Y}_t) = \sum_{j=1}^{N} p_{w_t}(x_{t+1}^{(i)} - f(x_t^{(j)}))p(x_t^{(j)}|\mathbb{Y}_t)\Delta^n, \tag{3.2b}$$

where the normalization factor γ_t is given by

$$\gamma_t = \sum_{i=1}^{N} p_{e_t}(y_t - h(x_t^{(i)}))p(x_t^{(i)}|\mathbb{Y}_{t-1})\Delta^n. \tag{3.3}$$

The mean estimate is approximated as

$$\hat{x}_{t|t} = \mathbb{E}\,(x_t|\mathbb{Y}_t) \approx \sum_{i=1}^{N} x_t^{(i)} p(x_t^{(i)}|\mathbb{Y}_t)\Delta^n. \tag{3.4}$$

In [12, 98], a nonlinear and non-Gaussian two-dimensional navigation estimation problem is solved using a grid-based integration of the Bayesian equations. Performance analysis in terms of the Cramér-Rao lower bound is presented in [12, 13]. Several implementation issues and how to use an adaptive grid size are also discussed in [12]. In [1], the navigation system is analyzed with a higher order state space model, incorporating drift terms in the filter. To avoid a higher dimension grid-based implementation, which is troublesome, the particle filter method is used.

In [99], the nonlinear and non-Gaussian problem is analyzed using grid-based integration as well as Monte Carlo integration for prediction and smoothing. Several nonlinear systems are compared using different techniques.

3.1.2 Stochastic Integration

Instead of using a deterministic numerical integration method, integrals can often be more efficiently solved using stochastic integration or the related Monte Carlo

method. Here, the focus is on the method denoted *importance sampling* (IS), [91], since it constitutes an important part of the particle filter. Many more methods exist, see for instance [13, 35, 49, 91]. The problem consists of approximating the expected mean of a function, using the underlying probability density $p(\cdot)$. The main idea is to use a proposal density, $q(\cdot)$, from which samples can easily be produced. Using this proposed density, the expected mean of an arbitrary function, $g(\cdot)$, can be written as

$$\mathbb{E}\left(g(X)\right) = \int g(x)p(x)\,dx = \int g(x)\frac{p(x)}{q(x)}q(x)\,dx. \tag{3.5}$$

Hence, the mean value of $g(x)$ is calculated by computing the mean of $g(x)p(x)/q(x)$ with samples from the density $q(x)$. Approximating the integral and calculating the mean value gives

$$\mathbb{E}\left(g(X)\right) \approx \frac{1}{N}\sum_{i=1}^{N}\underbrace{\frac{p(x^{(i)})}{q(x^{(i)})}}_{\gamma^{(i)}}g(x^{(i)}), \tag{3.6}$$

where $\{x_t^{(i)}\}_{i=1}^N$ are *independent identically distributed* (i.i.d.) samples from $q(\cdot)$ and the *importance weights* are defined as $\gamma^{(i)} = p(x^{(i)})/q(x^{(i)})$. In an example in [91, p. 80], it is shown that it may actually be favorable to generate samples from a pdf other than that of interest.

A brief theoretical motivation is given for the case of perfect sampling. Assume N independent samples drawn from a pdf $p(x)$. Define

$$\hat{g}_{_N} = \frac{1}{N}\sum_{i=1}^{N} g(x^{(i)}), \tag{3.7}$$

as an estimate of the mean of a function $g(x)$, with

$$I = \mathbb{E}\left(g(x)\right) = \int_{\mathbb{R}^n} g(x)p(x)\,dx. \tag{3.8}$$

The estimate is asymptotically unbiased and the sum converges almost surely to the true value

$$\text{Prob}\left(\lim_{N\to\infty}\hat{g}_{_N} = I\right) = 1, \tag{3.9}$$

by the *strong law of large numbers*. Moreover, [34], if

$$\sigma^2 = \text{Var}\left(g(x)\right) = \int_{\mathbb{R}^n}(g(x)-I)^2p(x)\,dx = \int_{\mathbb{R}^n} g^2(x)p(x)\,dx - I^2 < +\infty, \tag{3.10}$$

then by the *central limit theorem* the approximation error converges in distribution, that is

$$\lim_{N\to\infty}\sqrt{N}(\hat{g}_{_N} - I) \sim \mathcal{N}(0,\sigma^2). \tag{3.11}$$

The error of the estimate of the stochastic integration, $e = \hat{g}_N - I$, is of the order $\mathcal{O}(N^{-1/2})$, so the rate of convergence is independent of the state dimension, see for instance [13, 24, 89]. However, note that the constant in the convergence increases with the state dimension.

The particle filter method in Section 3.2 approximates the pdf from (2.10), using importance sampling with a certain proposal density. The main idea is that the sampled pdf will approach the true pdf if the number of samples is large enough.

3.2 Particle Filters

In estimation problems the task is to estimate unknown quantities from noisy observations, often with prior knowledge available. Therefore, it is natural to use a Bayesian approach. Many engineering problems are by nature recursive and require on-line solutions. For linear systems with a Gaussian noise assumption it is possible to derive a finite dimensional solution for the estimate. This recursive expression is given by the famous Kalman filter, [4, 47, 48]. For partially observed linear systems, the *hidden Markov model* (HMM), gives the solution. For many practical problems, linear models or the assumption of Gaussian noise, are not plausible. Therefore, there is a need for accurate recursive state estimation techniques for nonlinear and non-Gaussian problems. Monte Carlo techniques have been a growing research field lately due to improved computer performance. The seminal paper [36] marks the onset of a rebirth for algorithms based on sequential Monte Carlo simulation techniques for solving the Bayesian estimation problem. However, similar ideas have been discussed in [39, 40], where the conditional mean and covariance were calculated using importance sampling for recursive Bayesian estimation.

Sequential Monte Carlo methods, or particle filters, [26, 36, 89], provide general solutions to many problems, where linearizations and Gaussian approximations are intractable or would yield too low performance. Non-Gaussian noise assumptions and incorporation of constraints on the state variables can also be performed in a natural way.

Several versions of the particle filter exist. In Section 3.2.1, the original *sampling importance resampling* (SIR), [36], particle filter is presented and in Section 3.2.2, the *sequential importance sampling* (SIS) method is described. The *auxiliary particle filter* (APF), [86], presented in Section 3.2.3, is a candidate for heavy tailed noise or very informative measurements. If a linear-Gaussian substructure is present, the *marginalized particle filter* (MPF) or Rao-Blackwellized PF [6, 19, 20, 27, 28, 84, 93], can be used, as discussed in Section 3.2.4. All these particle filters are used in Part II, both for increased performance as well as for reduction of the computational complexity.

3.2.1 Sampling Importance Resampling

The particle filter theory presented in this section is inspired by [13, 27, 86, 89]. Consider the nonlinear discrete-time system from (2.1). The nonlinear prediction

density $p(x_{t+1}|\mathbb{Y}_t)$ and filtering density $p(x_t|\mathbb{Y}_t)$ for the Bayesian inference,[44], are given in (2.10), and repeated below for convenience.

$$p(x_{t+1}|\mathbb{Y}_t) = \int_{\mathbb{R}^n} p(x_{t+1}|x_t)p(x_t|\mathbb{Y}_t)\,dx_t, \tag{3.12a}$$

$$p(x_t|\mathbb{Y}_t) = \frac{p(y_t|x_t)p(x_t|\mathbb{Y}_{t-1})}{p(y_t|\mathbb{Y}_{t-1})}. \tag{3.12b}$$

The particle filter provides an approximative solution to the discrete-time recursive Bayesian estimation problem by updating an approximative description of the posterior filtering density. The particle filter approximates the probability density $p(x_t|\mathbb{Y}_t)$ by a large set of N particles $\{x_t^{(i)}\}_{i=1}^N$, where each particle has an assigned relative weight, $\gamma_t^{(i)}$, such that all weights sum to unity. The location and weight of each particle reflect the value of the density in that region of the state space. The particle filter updates the particle locations and the corresponding weights recursively with each new observation.

Often the normalization factor in (3.12b), $p(y_t|\mathbb{Y}_{t-1})$, is unknown. However, in the methods presented this factor is not necessary, since it is sufficient to evaluate

$$p(x_t|\mathbb{Y}_t) \propto p(y_t|x_t)p(x_t|\mathbb{Y}_{t-1}), \tag{3.13}$$

where the likelihood $p(y_t|x_t)$ is calculated from (2.1) using the known measurement noise density p_{e_t} as described in Section 2.2.

The main idea in the particle filter is to approximate $p(x_t|\mathbb{Y}_{t-1})$ with samples, according to

$$p(x_t|\mathbb{Y}_{t-1}) \approx \frac{1}{N}\sum_{i=1}^{N} \delta(x_t - x_t^{(i)}), \tag{3.14}$$

where δ is the delta-Dirac function. Inserting (3.14) into (3.13) yields a density to sample from. Details are given in Section 3.2.2. This is the original estimation idea. However, this approach leads to divergence, where almost all of the particles have zero weight. By introducing a selection or resampling step as proposed in [36] this can be handled. Mainly due to the resampling step and the increased computer capacity, there has lately been an increased research activity in the sequential Monte Carlo field. The resampling idea, from [36], is often referred to as *Bayesian bootstrap* or *sampling importance resampling* (SIR) and the algorithm is given in Algorithm 3.1. Note that it is possible and sometimes even preferable to generate more samples in the state update (step 4), and then resample back to N particles.

The point-estimate and the uncertainty region for the particle filter can be

Algorithm 3.1 Sampling Importance Resampling (SIR)

1: Set $t = 0$, generate N samples $\{x_0^{(i)}\}_{i=1}^N$ from the initial distribution $p_{x_0}(x_0)$.
2: Compute the weights $\gamma_t^{(i)} = p(y_t|x_t^{(i)})$ and normalize, i.e., $\tilde{\gamma}_t^{(i)} = \gamma_t^{(i)} / \sum_{j=1}^N \gamma_t^{(j)}, i = 1, \ldots, N$.
3: Generate a new set $\{x_t^{(i\star)}\}_{i=1}^N$ by resampling with replacement N times from $\{x_t^{(i)}\}_{i=1}^N$, where $\text{Prob}(x_t^{(i\star)} = x_t^{(j)}) = \tilde{\gamma}_t^{(j)}$.
4: Predict (simulate) new particles, i.e., $x_{t+1}^{(i)} = f(x_t^{(i\star)}, w_t^{(i)}), i = 1, \ldots, N$, where $w_t^{(i)}$ is drawn from the process noise with pdf $p_{w_t}(w_t)$.
5: Increase t and continue from step 2.

calculated as

$$\hat{x}_{t|t}^{\text{MMS}} = \mathbb{E}\,(x_t|\mathbb{Y}_t) = \int x_t\, p(x_t|\mathbb{Y}_t)\, dx_t \approx \sum_{i=1}^N \gamma_t^{(i)} x_t^{(i)}, \tag{3.15}$$

$$\begin{aligned} P_{t|t} &= \int (x_t - \hat{x}_{t|t}^{\text{MMS}})(x_t - \hat{x}_{t|t}^{\text{MMS}})^T p(x_t|\mathbb{Y}_t)\, dx_t \\ &\approx \sum_{i=1}^N \gamma_t^{(i)} (x_t^{(i)} - \hat{x}_{t|t}^{\text{MMS}})(x_t^{(i)} - \hat{x}_{t|t}^{\text{MMS}})^T. \end{aligned} \tag{3.16}$$

There are of course alternatives to the MMS point-estimate. One such candidate is to use a MAP-estimate, where the point-estimate is calculated using the particle with highest weight.

In many practical applications the system presented in (2.1) can be simplified. A common model uses additive noise with known pdfs, p_{w_t} and p_{e_t} according to

$$x_{t+1} = f(x_t) + w_t, \tag{3.17a}$$

$$y_t = h(x_t) + e_t. \tag{3.17b}$$

This structure simplifies the evaluation of the particle filter. The time update and the calculation of importance weights are given in (3.18) for each particle $i = 1, \ldots, N$.

$$x_{t+1}^{(i)} = f(x_t^{(i\star)}) + w_t^{(i)}, \tag{3.18a}$$

$$\gamma_t^{(i)} = p(y_t|x_t^{(i)}) = p_{e_t}(y_t - h(x_t^{(i)})). \tag{3.18b}$$

If the noise is not additive Theorem 2.1 may be applicable. As seen in (3.18b) the importance weights are sensitive to extreme outliers. If the possibility of outliers is not reflected in the probability density used to calculate the weights, there is a risk of ending up with all weights close to zero. To avoid this, these measurements can be discarded. In Example 3.1, the time-update (step 4) in the SIR method is demonstrated.

Example 3.1 (Particle filter pdf) *In Figure 3.1, the initial particle cloud from a Gaussian prior together with the importance weights are shown. Using the state equation for the time update, the particle cloud is spread out according to the figure.*

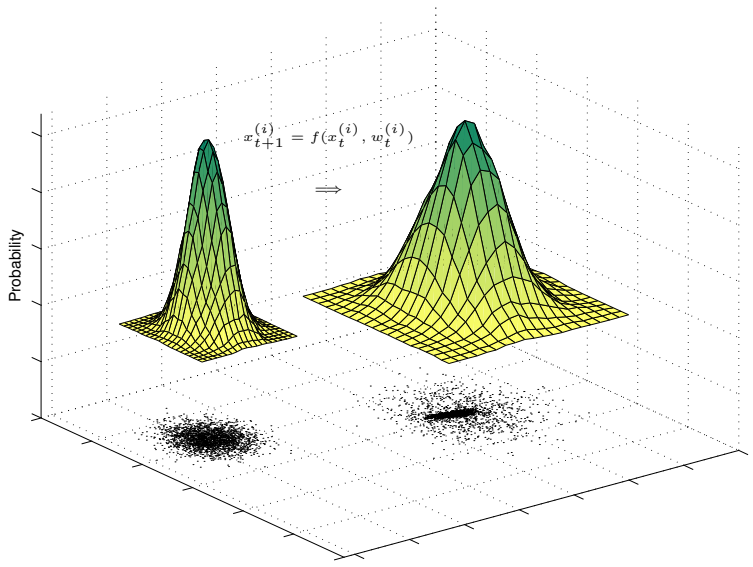


Figure 3.1: *The filtering pdf $p(x_t|\mathbb{Y}_t)$ (left) and the predicted pdf $p(x_{t+1}|\mathbb{Y}_t)$ (right) calculated from the particle cloud after step 4 in the SIR algorithm.*

The computational burden depends on the number of particles and on the resampling, which is a bottle-neck when it comes to parallelization. Fortunately, the resampling step can be efficiently implemented using a classical algorithm for sampling N ordered i.i.d. variables, [13, 27, 88]. The i.i.d sampling method is presented in Algorithm 3.2.

Algorithm 3.2 Sampling of ordered $\mathcal{U}(0,1)$ variables, [88, p. 96]

1: Sample $\bar{u}^{(i)} \sim \mathcal{U}(0,1)$, for $i = 1, \ldots, N$.
2: Set $u^{(N)} = \sqrt[N]{\bar{u}^{(N)}}$.
3: Compute $u^{(i)} = u^{(i+1)}\sqrt[i]{\bar{u}^{(i)}}$, for $i = N-1, \ldots, 1$.

In [13, p. 128], a very compact MATLAB implementation of Algorithm 3.2 is given. The resampling method is illustrated in Figure 3.2. The main idea is that particles

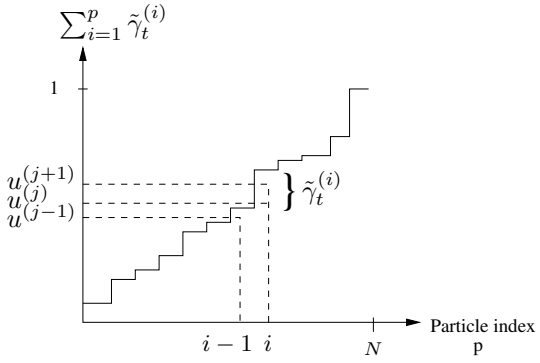


Figure 3.2: *A graphical interpretation of the particle filter resampling method. Assume N ordered uniform variables $u^{(k)}, k = 1, \ldots, N$. The normalized importance weight $\tilde{\gamma}_t^{(i)}$ for particle i has a value that in this case is large enough to include two samples of the ordered uniform variables. Hence, the corresponding particle $x_t^{(i)}$ is duplicated.*

with small weights are likely to be discarded and particles with large weights are copied, where the number of copies reflects the probability of the particle. This can be done by iterating over the ordered uniform samples $u^{(j)}, \quad j = 1, \ldots, N$ and comparing the cumulative sum of importance weights up to the current index. For example, in Figure 3.2 particle $x_t^{(i)}$ is duplicated. Another approach is to utilize deterministic resampling, e.g., to calculate the number of particles to be copied by using $N^{(i)} = \lfloor N\tilde{\gamma}_t^{(i)} \rfloor$, which could be somewhat faster. Also, instead of using Algorithm 3.2 an efficient sort function can be used. In [41], four different resampling schemes are compared with respect to their computational complexity and performance.

The main idea behind particle filtering seems rather straight forward and simple. However, one reason why it has been successful is that the empirical density, built up by the samples, converges to the true density if the number of particles is large enough. The fact that the error is independent of the state dimension makes the particle filter tractable for high dimensional problems, whereas for deterministic integration methods, the error is dependent on the grid-size. Further discussions are given in [22, 27], where convergence results are discussed. The upper bound on the variance of the estimation error is $c \cdot \mathcal{O}(N^{-1})$, where c is a constant. Note that this constant depends on the state dimension. In [89], this is discussed further in terms of the paper by [23], where it is shown that the dependence of the state dimension is basically linear. Another important issue is how many samples that are needed. The aim in [17] is to calculate bounds for the number of samples needed.

For many applications using the particle filter, depletion or sample impoverish-

ment may occur, i.e., the effective number of samples is reduced. This means that the particle cloud will not reflect the true density, since only a few of the particles will contribute to the approximation of the density. Several different methods are proposed in the literature to reduce this problem. By introducing an additional noise to the samples the depletion problem can be reduced. This technique is called *jittering* in [30], but a similar approach was introduced in [36] under the name *roughening*. In [28], the depletion problem is handled by introducing an additional *Markov Chain Monte Carlo* (MCMC) step to separate the samples.

3.2.2 Sequential Importance Sampling

The particle filter presented in Section 3.2.1 is based on importance sampling. In this section the recursive update is derived. This leads to a particle filter that is often referred to as *sequential importance sampling* (SIS), [27].

Recall the IS method in Section 3.1.2. Consider the following notation $\mathbb{X}_t$ representing the set of state vectors for different times up to and including t, that is $\mathbb{X}_t = \{x_0, x_1, \ldots, x_t\}$. Recall the IS-method in Section 3.1.2. If the samples $\mathbb{X}_t^{(i)}$ were drawn from the proposal density $q(\mathbb{X}_t|\mathbb{Y}_t)$, then the importance weights can be calculated as

$$\gamma_t^{(i)} \propto \frac{p(\mathbb{X}_t^{(i)}|\mathbb{Y}_t)}{q(\mathbb{X}_t^{(i)}|\mathbb{Y}_t)}. \tag{3.19}$$

The complete posterior density can then be rewritten using Bayes' theorem, the definition of conditional probability, and the Markov property inherent in the state space model.

$$\begin{aligned} p(\mathbb{X}_t|\mathbb{Y}_t) &= \frac{p(y_t|\mathbb{X}_t, \mathbb{Y}_{t-1})p(\mathbb{X}_t|\mathbb{Y}_{t-1})}{p(y_t|\mathbb{Y}_{t-1})} \\ &= \frac{p(y_t|x_t)p(x_t|\mathbb{X}_{t-1}, \mathbb{Y}_{t-1})p(\mathbb{X}_{t-1}|\mathbb{Y}_{t-1})}{p(y_t|\mathbb{Y}_{t-1})} \\ &= \frac{p(y_t|x_t)p(x_t|x_{t-1})}{p(y_t|\mathbb{Y}_{t-1})} p(\mathbb{X}_{t-1}|\mathbb{Y}_{t-1}). \end{aligned} \tag{3.20}$$

Ignoring the normalization factor

$$p(\mathbb{X}_t|\mathbb{Y}_t) \propto p(y_t|x_t)p(x_t|x_{t-1})p(\mathbb{X}_{t-1}|\mathbb{Y}_{t-1}). \tag{3.21}$$

Assume that the proposal density is chosen such that

$$q(\mathbb{X}_t|\mathbb{Y}_t) = q(x_t|\mathbb{X}_{t-1}, \mathbb{Y}_t)q(\mathbb{X}_{t-1}|\mathbb{Y}_{t-1}). \tag{3.22}$$

Inserting (3.21) and (3.22) into (3.19), the weights are recursively updated as

$$\gamma_t^{(i)} \propto \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})p(\mathbb{X}_{t-1}^{(i)}|\mathbb{Y}_{t-1})}{q(x_t^{(i)}|\mathbb{X}_{t-1}^{(i)}, \mathbb{Y}_t)q(\mathbb{X}_{t-1}^{(i)}|\mathbb{Y}_{t-1})} = \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|\mathbb{X}_{t-1}^{(i)}, \mathbb{Y}_t)}\gamma_{t-1}^{(i)}. \tag{3.23}$$

Particularly, the choice of $q(x_t^{(i)}|\mathbb{X}_{t-1}^{(i)}, \mathbb{Y}_t) = p(x_t^{(i)}|x_{t-1}^{(i)})$, gives the following update

$$\gamma_t^{(i)} \propto p(y_t|x_t^{(i)})\, \gamma_{t-1}^{(i)}. \tag{3.24}$$

Different particle filter methods use a selection or resampling step to avoid divergence. As pointed out in [27] the variance of the importance weights can only increase over time. Therefore, if nothing is done to adjust the particle cloud, it is impossible to avoid divergence or that the empirical density does not reflect the true one. The choice of resampling is often done by using a predefined criterion. An often suggested method is to study the *effective sample size*, N_{eff}, [13, 27, 68, 77]. The method relies on the calculation of how many samples in the particle cloud that actually contribute to the support of the probability density approximation. It is impossible to evaluate the expression analytically for N_{eff}, but an approximation is given by

$$\hat{N}_{\text{eff}} \approx \frac{1}{\sum_{i=1}^{N} (\gamma_t^{(i)})^2}. \tag{3.25}$$

If the effective number of samples is less than a predefined threshold, i.e., $\hat{N}_{\text{eff}} < N_{th}$, resampling should be applied. Note that, $1 \leq \hat{N}_{\text{eff}} \leq N$, where the upper bound is attained when all particles have the same weight and the lower bound when all probability mass is at one particle. Sometimes the resampling step in the particle filter presented in Section 3.2.2 is omitted and just imposed when it is needed to avoid divergence in the filter. The SIS method with resampling is summarized in Algorithm 3.3. A related method was originally developed in [78]. Note that the SIR method can be interpreted as SIS, where resampling is always performed. In that case all weights are set equal.

Algorithm 3.3 Sequential Importance Sampling (SIS), [27]

1: Set $t = 0$, generate N samples $\{x_0^{(i)}\}_{i=1}^N$ from the initial distribution $p_{x_0}(x_0)$. Initialize the importance weights $\tilde{\gamma}_{-1}^{(i)} = 1/N, \quad i = 1, \ldots, N$.
2: Compute the importance weights $\gamma_t^{(i)} = p(y_t|x_t^{(i)})\tilde{\gamma}_{t-1}^{(i)}$ and normalize, i.e., $\tilde{\gamma}_t^{(i)} = \gamma_t^{(i)} / \sum_{j=1}^N \gamma_t^{(j)},\ i = 1, \ldots, N$.
3: If resampling is applied, then generate a new set $\{x_t^{(i\star)}\}_{i=1}^N$ by resampling with replacement N times from $\{x_t^{(i)}\}_{i=1}^N$, with probability $\text{Prob}(x_t^{(i\star)} = x_t^{(i)}) = \tilde{\gamma}_t^{(i)}$ and set $\gamma_t^{(i)} = \frac{1}{N}$; otherwise let $\{x_t^{(i\star)}\} = \{x_t^{(i)}\}$, $i = 1, \ldots, N$.
4: Predict (simulate) new particles, i.e., $x_{t+1}^{(i)} = f(x_t^{(i\star)}, w_t^{(i)}),\ i = 1, \ldots, N$, where $w_t^{(i)}$ is drawn from the process noise with pdf $p_{w_t}(w_t)$.
5: Increase t and continue from step 2.

3.2.3 The Auxiliary Particle Filter

In [86], the *auxiliary particle filter* (APF) was proposed as an alternative particle filtering method. The idea is to increase the influence of particles with a large future likelihood, assuming that the estimate is not needed until after the next measurement. This is done by using an extra index to each particle, so the origin of the particle can be traced, while the likelihood at the next time step is evaluated. The effect is that those particles that were successful can be re-simulated and therefore the probability that the particle cloud moves in the desired direction is increased.

The APF extends the state x_t by predicting the state conditional upon particle k. At time t, the particle set $\{x_t^{(i)}\}_{i=1}^N$ and the corresponding weights $\tilde{\gamma}_t^{(i)}$ form the following approximations of the prediction and the filter densities

$$p(x_{t+1}|\mathbb{Y}_t) = \sum_{i=1}^{N} p(x_{t+1}|x_t^{(i)})p(x_t^{(i)}|\mathbb{Y}_t), \tag{3.26a}$$

$$p(x_{t+1}|\mathbb{Y}_{t+1}) \propto p(y_{t+1}|x_{t+1}) \sum_{i=1}^{N} p(x_{t+1}|x_t^{(i)})p(x_t^{(i)}|\mathbb{Y}_t), \tag{3.26b}$$

where $\mathbb{Y}_t$ is the set of cumulative measurements up to and including time t. By defining

$$p(x_{t+1}, k|\mathbb{Y}_{t+1}) \propto p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t^{(k)})p(k|\mathbb{Y}_t), \quad k = 1, \ldots, N, \tag{3.27}$$

samples can be drawn from this joint density and then discard the index, to produce a sample from the empirical filtering density as required. The index k is referred to as an auxiliary variable. Consider the joint density of particle k at time t and the state at time $t+1$. Bayes' theorem gives

$$\begin{aligned} p(x_{t+1}, k|\mathbb{Y}_{t+1}) &\propto p(y_{t+1}|x_{t+1}, k)p(x_{t+1}, k|\mathbb{Y}_t) = p(y_{t+1}|x_{t+1})p(x_{t+1}|k, \mathbb{Y}_t) \\ &= p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t^{(k)})p(k|\mathbb{Y}_t) \\ &= p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t^{(k)})p(k|\mathbb{Y}_t). \end{aligned} \tag{3.28}$$

Approximating this expression by replacing x_{t+1} with the expected mean

$$\mu_{t+1}^{(k)} = \mathbb{E}\left(x_{t+1}|x_t^{(k)}\right),$$

in the first factor gives

$$p(x_{t+1}, k|\mathbb{Y}_{t+1}) \propto p(y_{t+1}|\mu_{t+1}^{(k)})p(x_{t+1}|x_t^{(k)})p(k|\mathbb{Y}_t). \tag{3.29}$$

Marginalization over x_{t+1} yields

$$p(k|\mathbb{Y}_{t+1}) \propto p(y_{t+1}|\mu_{t+1}^{(k)})p(k|\mathbb{Y}_t). \tag{3.30}$$

Algorithm 3.4 Auxiliary Particle Filter (APF), [86]

1: Set $t = 0$, and generate N samples $\{x_0^{(i)}\}_{i=1}^N$ from $p(x_0)$, set $\mu_t^{(k)} = x_t^{(k)}$, $\tilde{\gamma}_0^{(j)} = 1/N, \quad k^{(j)} = j, \ j = 1, \ldots, N$.
2: Compute $\mu_{t+1}^{(k)} = \mathbb{E}\left(x_{t+1}|x_t^{(k)}\right)$.
3: Generate new indices $k^{(j)}$ by sampling N times from $p(k|\mathbb{Y}_{t+1}) \propto \tilde{\gamma}_t^{(k)} p(y_{t+1}|\mu_{t+1}^{(k)})$ and predict (simulate) the particles, i.e., $x_{t+1}^{(j)} = f(x_t^{(k^{(j)})}, w_t^{(j)}), \ j = 1, \ldots, N$ with different noise realizations.
4: Compute the likelihood weights $\gamma_t^{(j)} = \frac{p(y_{t+1}|x_{t+1}^{(j)})}{p(y_{t+1}|\mu_{t+1}^{(k^{(j)})})}$ for $j = 1, \ldots, N$ and normalize, i.e., $\tilde{\gamma}_t^{(j)} = \gamma_t^{(j)} / \sum_{j=1}^M \gamma_t^{(j)}$.
5: Perform an optional resampling of the set $\{x_{t+1}^{(i)}\}_{i=1}^N$, using the probability weights. If resampling is chosen then reset $\tilde{\gamma}_t^{(j)} = \frac{1}{N}, j = 1, \ldots, N$.
6: Increase t and continue from step 2.

Sampling from the density (3.28) can now be performed by resampling with replacement from the set $\{x_t^{(i)}\}_{i=1}^N$, where the index is chosen proportional to (3.30). The resampled candidates are then predicted using the system model. To summarize, the APF algorithm is given in Algorithm 3.4. The APF method is discussed in Paper G for a maneuvering target tracking application.

3.2.4 The Marginalized Particle Filter

In Section 3.2.1, the pdf $p(x_t|\mathbb{Y}_t)$ was approximated recursively using the particle filter for the whole state vector x_t. However, if the system has a linear-Gaussian sub-structure, this can be exploited to obtain a more efficient estimator. In practice this is done by marginalizing out the linear variables from $p(x_t|\mathbb{Y}_t)$. Denote the linear states with x_t^l and the nonlinear states x_t^p, with $\mathbb{X}_t^p = \{x_i^p\}_{i=0}^t$ and $\mathbb{X}_t = \{x_t^k, \mathbb{X}_t^p\}$. Hence,

$$p(\mathbb{X}_t|\mathbb{Y}_t) = p(\mathbb{X}_t^p, x_t^k|\mathbb{Y}_t) = p(x_t^k|\mathbb{X}_t^p, \mathbb{Y}_t) p(\mathbb{X}_t^p|\mathbb{Y}_t), \tag{3.31}$$

where $p(x_t^k|\mathbb{X}_t^p, \mathbb{Y}_t)$ is given by a Kalman filter and where $p(\mathbb{X}_t^p|\mathbb{Y}_t)$ is given by a particle filter. This marginalization idea is sometimes referred to as Rao-Blackwellization, see e.g., [6, 7, 20, 26, 27, 84, 93, 94].

Several different model structures can be analyzed using marginalization, [93]. Here, linear dynamics with a nonlinear measurement relation is considered. Using the notation x_t^p for the states that are estimated using the particle filer and x_t^k for the states that are estimated using the Kalman filter, the model will thus be

$$x_{t+1}^p = A_t^p x_t^p + A_t^k x_t^k + w_t^p, \tag{3.32a}$$
$$x_{t+1}^k = F_t^p x_t^p + F_t^k x_t^k + w_t^k, \tag{3.32b}$$
$$y_t = h_t(x_t^p) + e_t, \tag{3.32c}$$

where

$$w_t^p \sim \mathcal{N}(0, Q_t^p), \tag{3.32d}$$
$$w_t^k \sim \mathcal{N}(0, Q_t^k), \tag{3.32e}$$
$$e_t \sim \mathcal{N}(0, R_t). \tag{3.32f}$$

All noise signals are considered white and independent. In Algorithm 3.5, the MPF is summarized for the model given in (3.32). For a detailed derivation of this algorithm the reader is referred to [94].

In Algorithm 3.5, the MPF is presented for white and independent noise. If for instance the process noise is correlated, which is common in many positioning and target tracking applications, the algorithm must be modified. If $\bar{w}_t = Gw_t$, then

$$\bar{Q}_t = \mathbb{E}\left(\bar{w}_t \bar{w}_t^T\right) = GQ_tG^T = \begin{pmatrix} Q_t^p & \mathcal{S}_t \\ \mathcal{S}_t^T & Q_t^k \end{pmatrix}. \tag{3.38}$$

In order to adjust for the correlated process noise, define the scalar product of two stochastic variables as $<u, v> = \mathbb{E}\left(uv^T\right)$ and the norm $\|u\| = \sqrt{<u, u>}$. According to [47] the noise can then be decorrelated by

$$\tilde{w}_t^k = \bar{w}_t^k - <\bar{w}_t^k, \bar{w}_t^p>\|\bar{w}_t^p\|^{-2}\bar{w}_t^p = \bar{w}_t^k - \mathcal{S}_t(Q_t^p)^{-1}\bar{w}_t^p, \tag{3.39}$$

where $\mathcal{S}_t = <\bar{w}_t^p, \bar{w}_t^k>$. Once Step 4 (b) in Algorithm 3.5 has been executed $\bar{w}_t^p$ is known and it is straightforward to calculate $\tilde{w}_t^k$, and the decorrelation is complete.

Note that several permutations are possible when selecting the partition of the state-vector, since the particle filter can include part of the x_t^k states. Two relevant aspects with respect to this partitioning are how it will affect the computational load and the estimation performance. This is discussed in Paper B. In Paper E, the MPF is used in a bearings-only target tracking application.

Algorithm 3.5 Marginalized Particle Filter (MPF)

1: Initialization: For $i = 1, \ldots, N$, initialize the particles, $x_{0|-1}^{p,(i)} \sim p_{x_0^p}(x_0^p)$ and set $\{x_{0|-1}^{k,(i)}, P_{0|-1}^{(i)}\} = \{\bar{x}_0^k, \bar{P}_0\}$. Let $t = 0$.

2: For $i = 1, \ldots, N$, evaluate the importance weights $\gamma_t^{(i)} = p(y_t|\mathbb{X}_t^{p,(i)}, \mathbb{Y}_{t-1})$ according to the likelihood

$$p(y_t|\mathbb{X}_t^p, \mathbb{Y}_{t-1}) = \mathcal{N}(h_t(x_t^p),\, R_t) \tag{3.33}$$

and normalize $\tilde{\gamma}_t^{(i)} = \frac{\gamma_t^{(i)}}{\sum_{j=1}^N \gamma_t^{(j)}}$.

3: PF measurement update: Resample N particles with replacement according to,

$$\text{Prob}(x_{t|t}^{p,(i)} = x_{t|t-1}^{p,(j)}) = \tilde{\gamma}_t^{(j)}. \tag{3.34}$$

4: PF time update and Kalman filter update

(a) Kalman filter measurement update,

$$\hat{x}_{t|t}^{k,(i)} = \hat{x}_{t|t-1}^{k,(i)},\; P_{t|t} = P_{t|t-1}. \tag{3.35}$$

(b) PF time update: For $i = 1, \ldots, N$,

$$x_{t+1|t}^{p,(i)} \sim p(x_{t+1|t}^p|\mathbb{X}_t^{p,(i)}, \mathbb{Y}_t), \tag{3.36}$$

where

$$p(x_{t+1}^{p,(i)}|\mathbb{X}_t^{p,(i)}, \mathbb{Y}_t) = \mathcal{N}(A_t x_t^{p,(i)} + A_t^k \hat{x}_{t|t}^{k,(i)},\; A_t^k P_{t|t}(A_t^k)^T + Q_t^p).$$

(c) Kalman filter time update,

$$\begin{aligned}
\hat{x}_{t+1|t}^{k,(i)} &= F_t^k \hat{x}_{t|t}^{k,(i)} + F_t^p x_t^{p,(i)} + L_t(x_{t+1|t}^{p,(i)} - A_t^p x_t^{p,(i)} - A_t^k \hat{x}_{t|t}^{k,(i)}),\\
P_{t+1|t} &= F_t^k P_{t|t}(F_t^k)^T + Q_t^k - L_t M_t L_t^T,\\
M_t &= A_t^k P_{t|t}(A_t^k)^T + Q_t^p,\\
L_t &= F_t^k P_{t|t}(A_t^k)^T M_t^{-1},
\end{aligned}$$

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

3.2.5 Other Particle Filter Methods

In this thesis the SIR, SIS, APF, and MPF versions of the particle filter are presented. However, there are other methods or variations that are of interest. In this section different topics related to resampling, multiple models and sub-optimal approximations are briefly presented.

Resampling is crucial for reducing the degeneracy problem for particle filters. Hence, a lot of work has been invested trying to improve it. However, the introduction of the resampling step leads to a loss of diversity among the particles. In [26], the *regularized particle filter* (RPF) was proposed as a solution to the problem. A summary of the RPF is given in [82]. Instead of drawing samples from a discrete approximation the samples are drawn from a continuous approximation of the density, where the posterior filtering density is constructed using kernel functions around the particles. The RPF uses the following approximation

$$p(x_t|\mathbb{Y}_t) \approx \sum_{i=1}^{N} \gamma_t^{(i)} \mathcal{K}_\alpha(x_t - x_t^{(i)}), \quad \text{with } \mathcal{K}_\alpha(x) = \frac{1}{\alpha^n}\mathcal{K}(x/\alpha), \tag{3.40}$$

where $\mathcal{K}(\cdot)$ is a *kernel function*, $x_t \in \mathbb{R}^n$ and $\alpha > 0$ is the *kernel bandwidth*. The kernel function is a symmetric pdf with zero mean and finite variance.

In [69, 70, 71], a particle filtering approach is presented, utilizing Gaussian or *Gaussian sum* (GS) approximations, [4, 96]. The *Gaussian particle filter* (GPF), [70], approximates the filtering and predictive pdfs by Gaussian densities using the particle filter methodology. In the *Gaussian sum particle filter* (GSPF), [71], the filtering and prediction densities are approximated by finite Gaussian mixtures, and the sequential update relies on a sampling based method. The idea is to approximate the pdf, for instance

$$p(x_t|\mathbb{Y}_t) \approx \sum_{i=1}^{M} \gamma_t^{(i)} \mathcal{N}(x_t; \mu_i, P_i), \tag{3.41}$$

where $M \ll N$ and the mean (μ_i) and covariance (P_i) are calculated from the particles.

Several methods involve topics related to multiple model estimation. The *multiple model particle filter*, [28, 83, 89], handles nonlinear filtering with switching dynamics. In Paper F, an example of this is given using the APF. The *jump Markov linear system* is discussed for a particle filter implementation in [25, 28] and jump Markov nonlinear systems are discussed in [29]. In [18], a multiple model particle filtering method for Markovian switching systems is presented, using an RPF for each mode. The *local linearized particle filter*, [89], or EKF-PF in [101], use an EKF for each particle to generate a Gaussian importance distribution. Similar, the *unscented particle filter* (UPF), [101], using the *unscented Kalman filter* (UKF), [45, 46] is adopted for particle filtering.

Usually, the PF is implemented in on-line applications, where data are processed sequentially in order of arrival. However, sometimes the estimate is not needed until

future measurements are available or the measurements are not ordered in time. Smoothing, i.e., improving estimation using non-causal filtering is discussed for the particle filter in [27, 28, 33, 67]. The *out-of-sequence measurement* problem, when measurements arrive in non consecutive order, is discussed for particle filtering in for instance [79, 80]. Several of the above modifications or algorithms are also presented in [26, 89].

4

Summary

The focus of this thesis is on recursive Bayesian estimation methods, where the particle filter is the main tool for solving the non-tractable Bayesian recursions. The *extended Kalman filter* (EKF) and linearized multiple models are classical methods used for comparison. In various positioning and tracking applications different aspects are emphasized. To summarize, the particle filter method is easy to implement for many nonlinear discrete-time systems. It can handle non-Gaussian noise and it is easy to impose constraints. For instance, the nonlinear measurement relation or the constraints on the state vector could be defined by a database, so no analytic relation exist. This makes a classical Kalman filter implementation troublesome. Performance issues are mostly discussed in terms of the *Cramér-Rao lower bound* (CRLB), numerically and analytically, depending on the system structure.

Theoretical results: In Paper A, quantization effects in the measurement relation are studied. The classical quantization theory is revisited and topics such as moment-based estimation, band-limited noise and dithering noise are emphasized. Static and dynamic estimation is discussed using maximum likelihood estimation and performance is calculated using the CRLB. The particle filter is adopted to the measurement quantization case and shown to be superior compared to a Kalman filter assuming *additive uniform noise*.

In Paper B, the computational complexity of the marginalized particle filter

is analyzed using the introduced *equivalent flop* measure. For the studied linear-Gaussian dynamics, the state vector can be partitioned in several ways, resulting in different computational complexity.

Positioning applications: In Paper C, a framework for positioning at sea using a distance measuring equipment together with chart information is presented based on particle filtering. An analytic performance bound using the CRLB is calculated for the system. In particular two applications are studied: an underwater navigation system based on sonar measurements and a depth map and a novel surface navigation system based on radar measurements compared to a sea chart.

In Paper D, the particle filter and the EKF are applied to an industrial robot, with joint flexibility. The aim is to improve positioning on the tool side using sensor fusion of motor side measurements together with accelerometer measurements at the end-effector. Estimation performance for a simplified dynamic model in the filter is compared to the CRLB. The aim is to use the estimates for iterative learning control or for position related hypothesis testing.

Tracking applications: In Paper E, several bearings-only applications are studied using the particle filter and the marginalized particle filter, where the *range parameterized extended Kalman filter*, is used as comparison. The tracking applications are: Air-to-Air, Air-to-Sea and Sea-to-Sea. Both extensive Monte Carlo simulations as well as experimental data are provided. Using a sea chart, interpreted as an external sensor, the estimation performance can be enhanced.

In Paper F, the data association problem for a target tracking application is studied. The aim is to incorporate classical association methods to the particle filter framework. An association algorithm for the particle filter based on the joint probabilistic data association method is proposed. In order to reduce computational complexity the idea of a controller for the number of particles is also introduced.

In Paper G, the auxiliary particle filter is adopted to the maneuvering target application, with hard constraints on system states. In a simulation study it is compared to the classical particle filter and the interacting multiple model.

In Paper H, a real-time collision avoidance algorithm based on the particle filter is proposed, using a forward looking radar and a braking algorithm based on hypothesis testing. An alternative EKF based approach, using stochastic integration in the hypothesis test, is also provided with comparable performance.

Bibliography

[1] M. Ahlström and M. Calais. Bayesian terrain navigation with Monte Carlo method. Master's Thesis LiTH-ISY-EX-3051, Department of Electrical Engineering. Linköping University, Linköping, Sweden, 2000. In Swedish.

[2] J. Aldrich. R. A. Fisher and the making of maximum likelihood 1912–1922. *Statistical Science*, 12(3):162–176, 1997.

[3] D. Alspach and H. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448, 1972.

[4] B.D.O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, NJ, 1979.

[5] P. Andersson. Adaptive forgetting in recursive identification through multiple models. *International Journal of Control*, 42(5):1175–1193, 1985.

[6] C. Andrieu and A. Doucet. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society*, 64(4):827–836, 2002.

[7] C. Andrieu and S. J. Godsill. A particle filter for model based audio source separation. In *ICA 2000*, Helsinki, Finland, June 2000.

[8] S. Arulampalam and B. Ristic. Comparison of the particle filter with range-parameterised and modified polar EKFs for angle-only tracking. In *Proceedings of SPIE, Signal and Data Processing of Small Target*, pages 288–299, Orlando, FL, USA, 2000.

[9] Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*, volume 179 of *Mathematics in Science and Engineering*. Academic Press, 1988.

[10] Y. Bar-Shalom and X. R. Li. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, 1993.

[11] T. R. Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53(370), 1763. Reprinted in *Biometrika*, 45(293), 1958.

[12] N. Bergman. *Bayesian Inference in Terrain Navigation*. Linköping Studies in Science and Technology. Licentiate Thesis No. 649, Linköping University, Linköping, Sweden, 1997.

[13] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Linköping Studies in Science and Technology. Dissertations No. 579, Linköping University, Linköping, Sweden, 1999.

[14] S. S. Blackman. *Multiple-Target Tracking with Radar Applications*. Artech House, Norwood, MA, 1986.

[15] S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[16] H.A.P. Blom. An efficient filter for abruptly changing systems. In *Proceedings of the 23rd IEEE Conference on Decision and Control*, pages 656–658, Las Vegas, NV, USA, 1984.

[17] Y. Boers. On the number of samples to be drawn in particle filtering. In *IEE Colloquium on Target Tracking: Algorithms and Applications*, pages 5/1–5/6, 1999.

[18] Y. Boers and J. N. Driessen. Interacting multiple model particle filter. *IEE Proceedings of Radar, Sonar and Navigation*, 150(5):344–349, October 2003.

[19] G. Casella and C. P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

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

[21] H. Cramér. *Mathematical Methods of Statistics*. Princeton, NJ: Princeton University Press, 1946.

[22] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50:736–746, March 2002.

[23] F. Daum and J. Huang. Curse of dimensionality and particle filters. In *Proceedings of IEEE Aerospace Conference*, 2003.

[24] P. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Academic Press, 1984.

[25] A. Doucet and C. Andrieu. Iterative algorithms for optimal state estimation of jump Markov linear systems. In *Proceedings IEEE Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 2487–2490, 1999.

[26] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

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

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

[29] H. Driessen and Y. Boers. An efficient particle filter for jump Markov nonlinear systems. In *IEE International Seminar on Target Tracking: Algorithms and Applications*, pages 19–22, March 2004.

[30] P. Fearnhead. *Sequential Monte Carlo Methods in Filter Theory*. PhD thesis, University of Oxford, 1998.

[31] R. A. Fisher. On an absolute criterion for fitting frequency curves. *Messenger of Mathematics*, 41:155–160, 1912.

[32] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London, Ser. A*, pages 309–368, 1922.

[33] W. Fong, J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing with application to audio signal enhancement. *IEEE Transactions on Signal Processing*, 50:438–449, February 2002.

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

[35] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1996.

[36] N. J. Gordon, D. J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[37] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons Ltd, 2000.

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

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

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

[41] J. Hol. Resampling in particle filters. Master's Thesis LiTH-ISY-EX-ET-0283-2004, Department of Electrical Engineering. Linköping University, Linköping, Sweden, 2004.

[42] A. Holtsberg. *A Statistical Analysis of Bearings-Only Tracking*. PhD thesis, Department of Mathematical Statistics, Lund University, 1992. LUTFD2/TFMS–1010.

[43] C. Hue, J.P. Le Cadre, and P. Pérez. Sequential Monte Carlo methods for target tracking and data fusion. In *IEEE Transactions on Signal Processing*, volume 50, pages 309–325, February 2002.

[44] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, 1970.

[45] S. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92:401–422, March 2004.

[46] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45:477–482, March 2000.

[47] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Information and System Sciences. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[48] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the AMSE–Journal of Basic Engineering*, 82:35–45, 1960.

[49] R. Karlsson. *Simulation Based Methods for Target Tracking*. Linköping Studies in Science and Technology. Licentiate Thesis No. 930, Linköping University, Linköping, Sweden, February 2002.

[50] R. Karlsson and N. Bergman. Auxiliary particle filters for tracking a maneuvering target. In *Proceedings of the 39:th IEEE Conference on Decision and Control*, pages 3891–3895, Sydney, Australia, December 2000.

[51] R. Karlsson and F. Gustafsson. Filtering and estimation for quantized sensor information. *Submitted to IEEE Transactions on Signal Processing.*

[52] R. Karlsson and F. Gustafsson. Recursive Bayesian estimation – bearings-only applications. *Accepted for publication in IEE Proceedings on Radar, Sonar, and Navigation. Special issue on target tracking: Algorithms and Applications.*

[53] R. Karlsson and F. Gustafsson. Surface and underwater navigation using particle filters. *Submitted to IEEE Transactions on Signal Processing.*

[54] R. Karlsson and F. Gustafsson. Monte Carlo data association for multiple target tracking. In *IEE International Seminar on Target Tracking: Algorithms and Applications*, pages 13/1–13/5, Enschede, The Netherlands, October 2001.

[55] R. Karlsson and F. Gustafsson. Range estimation using angle-only target tracking with particle filters. In *Proceedings of American Control Conference*, volume 5, pages 3743–3748, Arlington, Virginia, USA, June 2001. Invited paper.

[56] R. Karlsson and F. Gustafsson. Particle filter for underwater terrain navigation. In *IEEE Statistical Signal Processing Workshop*, pages 526–529, St. Louis, MO, USA, October 2003. Invited paper.

[57] R. Karlsson and F. Gustafsson. A system and method for surface navigation using radar and sea map. Swedish patent application. Patent pending SE-0400264-8, February 2004.

[58] R. Karlsson and F. Gustafsson. Particle filtering for quantized sensor information. In *Proceedings of the 13th European Signal Processing Conference*, Antalya, Turkey, September 2005. Submitted as invited paper.

[59] R. Karlsson and F. Gustafsson. Using the particle filter as mitigation to GPS vulnerability for navigation at sea. In *Proceedings of IEEE Statistical Signal Processing Workshop*, Bordeaux, France, July 2005. Submitted as invited paper.

[60] R. Karlsson, F. Gustafsson, and T. Karlsson. Particle filtering and Cramér-Rao lower bound for underwater navigation. In *Proceedings IEEE Conference on Acoustics, Speech and Signal Processing*, Hong Kong, April 2003.

[61] R. Karlsson, J. Jansson, and F. Gustafsson. Model-based statistical tracking and decision making for collision avoidance application. In *Proceedings of American Control Conference*, Boston, MA, USA, June 2004.

[62] R. Karlsson and M. Norrlöf. Bayesian state estimation of a flexible industrial robot. *Submitted to IEEE Transactions on Control Systems Technology.*

[63] R. Karlsson and M. Norrlöf. Bayesian position estimation of an industrial robot using multiple sensors. In *Proceedings of the IEEE Conference on Control Applications*, Taipei, Taiwan, September 2004.

[64] R. Karlsson and M. Norrlöf. Position estimation and modeling of a flexible industrial robot. In *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, July 2005. To appear.

[65] R. Karlsson, T. Schön, and F. Gustafsson. Complexity analysis of the marginalized particle filter. *To appear in IEEE Transactions on Signal Processing.*

[66] S. Kay. *Fundamentals of Statistical Signal Processing*. Prentice Hall, 1993.

[67] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphics Statistics*, 5(1):1–25, 1996.

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

[69] J. H. Kotecha and P. M. Djurić. Gaussian sum particle filtering for dynamic state space models. In *Proceedings IEEE Conference on Acoustics, Speech and Signal Processing*, volume 6, pages 3465–3468, Salt Lake City, UT, USA, 2001.

[70] J. H. Kotecha and P. M. Djurić. Gaussian particle filtering. *IEEE Transactions on Signal Processing*, 51:2592–2601, October 2003.

[71] J. H. Kotecha and P. M. Djurić. Gaussian sum particle filtering. *IEEE Transactions on Signal Processing*, 51:2602–2612, October 2003.

[72] S. C. Kramer and H. W. Sorenson. Bayesian parameter estimation. *IEEE Transactions on Automatic Control*, 33:217–222, 1988.

[73] T. R. Kronhamn. Bearings-only target motion analysis based on a multihypothesis Kalman filter and adaptive ownship motion control. In *IEE Proceedings on Radar, Sonar and Navigation*, volume 145, pages 247–252, 1998.

[74] E. L. Lehmann. *Theory of Point Estimation*. John Wiley and Sons, 1983.

[75] X. R. Li and V. P. Jilkov. A survey of maneuvering target tracking–part III: Measurement models. In *Proceedings of SPIE Conference on signal and data processing of small targets*, volume 4473, pages 423–446, July 2001.

[76] X. R. Li and V.P. Jilkov. A survey of maneuvering target tracking: Dynamics models. In *Proceedings of SPIE Conference on signal and data processing of small targets*, volume 4048, pages 212–235, April 2000.

[77] J. S. Liu. Metropolized independent sampling with comparison to rejection ampling and importance sampling. *Statistics and Computing*, 6:113–119, 1996.

[78] J. S. Liu and E. Chen. Blind deconvolation via sequential imputation. *Journal of American Statistical Association*, 90:567–576, 1995.

[79] M. Mallick, T. Kirubarajan, and S. Arulampalam. Out-of-sequence measurement processing for tracking ground target using particle filters. In *Proceedings of the IEEE Aerospace Conference*, volume 4, pages 1809—1818, Big Sky, MT, USA, March 2002.

[80] M. Mallick and A. Marrs. Comparison of the KF and particle filter based out-of-sequence measurement filtering algorithms. In *Proceedings of the Sixth International Conference on Information Fusion*, volume 1, pages 422–429, Cairns, Qld., Australia, July 2003.

[81] M. Mallick, S. Maskell, T. Kirubarajan, and N. Gordon. Littoral tracking using the particle filter. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 1, pages 935–942, Annapolis, MD, USA, July 2002.

[82] S. Maskell and N. Gordon. A tutorial on particle filter for on-line nonlinear/non-Gaussian Bayesian tracking. In *IEE Workshop. Target Tracking: Algorithms and Applications*, Enschede, The Netherlands, October 2001.

[83] S. McGinnity and G. Irwin. Multiple model estimation using the bootstrap filter. In *IEE Colloquium on Target Tracking and Data Fusion (Digest No. 1998/282)*, 1998.

[84] P-J. Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation*. Linköping Studies in Science and Technology. Licentiate Thesis No. 945, Linköping University, Linköping, Sweden, May 2002.

[85] N. Peach. Bearings-only tracking using a set of range-parameterised extended Kalman filters. In *IEE Proceedings of Control Theory and Applications*, volume 142, pages 73–80, January 1995.

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

[87] D. B. Reid. The application of multiple target tracking theory to ocean surveillance. In *Proceedings of the 18:th IEEE Conference on Decision and Control*, pages 1046–1052, Fort Lauderdale, FL, USA, 1979.

[88] B. D. Ripley. *Stochastic Simulation*. John Wiley, 1988.

[89] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[90] C. P. Robert. *The Bayesian Choice*. Springer-Verlag, second edition, 2001.

[91] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, 1999.

[92] P. N. Robinson and M. R. Yin. Modified spherical coordinates for radar. In *Proceedings AIAA Guidance, Navigation and Control Conference*, pages 55–64, August 1994.

[93] T. Schön. *On Computational Methods for Nonlinear Estimation*. Linköping Studies in Science and Technology. Licentiate Thesis No. 1047, Linköping University, Linköping, Sweden, October 2003.

[94] T. Schön, F. Gustafsson, and P-J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *To appear in IEEE Transactions on Signal Processing*.

[95] M. A. Simard and F. Begin. Central level fusion of radar and IRST contacts and the choice of coordinate system. In *SPIE Signal and Data Processing of Small Targets*, volume 1954, pages 462–472, Orlando, FL, USA, July 1993.

[96] H. W. Sorenson and D. L. Alspach. Recursive Bayesian estimation using Gaussian sums. *Automatica*, 7:465–479, 1971.

[97] D. V. Stallard. An angle-only tracking filter in modified sperical coordinates. In *Proceedings AIAA Guidance and Navigation and Control Conference*, pages 542–550, 1987.

[98] M. Svensson. Implementation and evaluation of Bayesian terrain navigation. Master's Thesis LiTH-ISY-EX-2039, Department of Electrical Engineering. Linköping University, Linköping, Sweden, 1999. In Swedish.

[99] H. Tanizaki. Nonlinear and nonnormal filters using Monte Carlo methods. *Computational Statistics and Data Analysis*, 25:417–439, 1997.

[100] P. Tichavsky, P. Muravchik, and A. Nehorai. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing*, 46(5):1386–1396, 1998.

[101] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. Technical Report CUED/F-INFENG/TR 380, Cambridge University Department of Engineering, Cambridge, UK, May 2000.

[102] H. L. Van Trees. *Detection, Estimation and Modulation Theory*. Wiley, New York, 1968.

[103] B. Widrow. A study of rough amplitude quantization by means of Nyquist sampling theory. *IRE Transactions on Circuit Theory*, pages 266–276, December 1956.

[104] B. Widrow, I. Kollar, and M-C Liu. Statistical theory of quantization. *IEEE Transactions on Instrumentation and Measurement*, pages 353–361, April 1996.

Part II

Publications

Paper A

Filtering and Estimation for Quantized Sensor Information

Edited version of the paper:

> R. Karlsson and F. Gustafsson. Filtering and estimation for quantized sensor information. *Submitted to IEEE Transactions on Signal Processing.*

Parts of the paper in:

> R. Karlsson and F. Gustafsson. Particle filtering for quantized sensor information. In *Proceedings of the 13th European Signal Processing Conference*, Antalya, Turkey, September 2005. Submitted as invited paper.

Preliminary version published as Technical Report LiTH-ISY-R-2674, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Filtering and Estimation for Quantized Sensor Information

Rickard Karlsson and Fredrik Gustafsson,

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

Abstract

The implication of quantized sensor information on estimation and filtering problems is studied. The close relation between sampling and quantization theory was earlier reported by Widrow, Kollar and Liu (1996). They proved that perfect reconstruction of the probability density function (pdf) is possible if the characteristic function of the sensor noise pdf is band-limited. These relations are here extended by providing a class of band-limited pdfs, and it is shown that adding such dithering noise is similar to anti-alias filtering in sampling theory. This is followed up by the implications for Maximum Likelihood and Bayesian estimation. The Cramér-Rao lower bound (CRLB) is derived for estimation and filtering on quantized data. A particle filter (PF) algorithm that approximates the optimal nonlinear filter is provided, and numerical experiments show that the PF attains the CRLB, while second-order optimal Kalman filter approaches can perform quite bad.

Keywords: Quantization, Estimation, Filtering, Cramér-Rao Lower Bound.

1 Introduction

Quantization was a well studied topic in *digital signal processing* (DSP) some decades ago [15], when the underlying reason was the finite computation precision in micro-processors. Today, new reasons have appeared that motivate a revisit of the area:

- Cheap low-quality sensors have appeared on the market and in many consumer products, which open up many new application areas for embedded DSP algorithms, where the sensor resolution is much less than the micro-processor resolution.

- The increased use of distributed sensors in communication networks with limited bandwidth.

- Some sensors are naturally quantized as radar range, vision devices, cogged wheels to measure angular speeds *etc.*. With increased performance requirements, quantization effects become important to analyze.

- The renewed interest in nonlinear filtering with the advent of the particle filter [6] enables a tool to take quantization effects into account in the filter design.

In these cases, one can regard the sensor readings as quantized. All sub-sequent computations are done with floating point precision, or in fixed-point arithmetics with adaptive scaling of all numbers, which means that internal quantization effects can be neglected. Thus, the quantization effects studied in this paper differ from the ones studied decades ago [15].

The first contribution of this paper is to revisit classical quantization theory from [22, 23]. They show that quantization adds two kind of errors to the measurement, the first one is a direct effect that can be modeled as *additive uniform noise* (AUN), and the other one is an intrinsic alias like uncertainty, where fast variations in the *probability density function* (pdf) of the measurement noise are folded to low frequencies. The theory in [23] is extended by discussing the role of dithering noise as a remedy to this folding effect similar to anti-alias filtering.

The second aim of this paper is to analyze the influence of quantization effects on the following estimation and filtering problems:

1. Estimate the parameter x in the quantized measurement $y_i = \mathcal{Q}(x + e_i)$, where e_i is measurement noise and $\mathcal{Q}(\cdot)$ denotes the quantization operator.

2. Estimate the parameter x in the (nonlinear) least squares model $h(x)$ using quantized measurement $y_i = \mathcal{Q}(h(x) + e_i)$.

3. Estimate the state x_i in the (nonlinear) dynamic model $x_{i+1} = f(x_i, w_i)$ using quantized measurement $y_i = \mathcal{Q}(h(x_i) + e_i)$, where w_i is process noise.

It will be described how to modify moment-based, likelihood-based and Bayesian approaches to quantized information, and here the result on reconstruction and anti-alias dithering are instrumental.

The paper is organized as follows: In Section 2, quantization as area sampling is revisited. Section 3 presents some new results on band-limited noise and the anti-alias equivalent for quantization is given. In Section 4, moment-based parameter estimation is discussed. In Section 5, ML-estimation for different quantization cases are presented and information bounds in terms of the CRLB are derived. In Section 6, the particle filer is applied to quantized sensor information for a dynamics system. In Section 7, the conclusions are given.

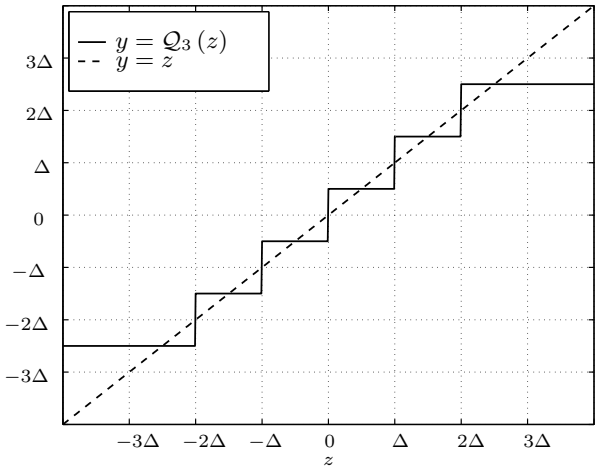


Figure 1: *Uniform quantization using a midriser quantizer with quantization step* Δ*. The quantized set is given by* $y = \mathcal{Q}_3(z) = \{-m\Delta+\frac{\Delta}{2}, \ldots, (m-1)\Delta + \frac{\Delta}{2}\}$*, for* $m = 3$*, i.e.,* $2m = 6$ *quantization levels.*

2 Fundamental Properties of Quantized Noise

In this paper, the quantization function is restricted to the case of uniform amplitude quantization. In principle, it is implemented either as the *midtread* or the *midriser* quantizer, as described in [14]. If not saturated these are given as:

$$\mathcal{Q}_m(z) = \Delta \left\lfloor \frac{z}{\Delta} + \frac{1}{2} \right\rfloor, \text{or} \tag{1a}$$

$$\mathcal{Q}_m(z) = \Delta \left\lfloor \frac{z}{\Delta} \right\rfloor + \frac{\Delta}{2}, \tag{1b}$$

respectively. Here, $\mathcal{Q}_m(\cdot)$ denotes the nonlinear quantization mapping. The $\lfloor \cdot \rfloor$ operator rounds downwards to the nearest integer. To keep a unified notation with the sign quantization $\mathcal{Q}_1(z) = \text{sign}(z)$, the midriser convention will be used, so $y \in \{-m\Delta+\frac{\Delta}{2}, \ldots, (m-1)\Delta+\frac{\Delta}{2}\}$, with $\Delta = 2^{-b}$, using b bits, $2m = 2^b$ levels and $2^b - 1$ thresholds, as illustrated in Figure 1. The sign quantization corresponds to $b = 1$, $m = 1$ and $\Delta = 2$ in this notation. That is, the measurement is defined as

$$y = \begin{cases} -m\Delta + \frac{\Delta}{2}, & z < m\Delta, \\ \Delta \left\lfloor \frac{z}{\Delta} \right\rfloor + \frac{\Delta}{2}, & m\Delta < z \leq m\Delta, \\ m\Delta - \frac{\Delta}{2}, & z \geq m\Delta. \end{cases} \tag{2}$$

The distribution of y differ from the distribution of z for three reasons:

1. Saturation effects when $|z| > m\Delta$.
2. The direct quantization effect that can be modeled as *additive uniform noise* (AUN) with variance $\Delta^2/12$.
3. Intricate pdf aliasing effects.

Saturation will be ignored for the rest of this section, where the goal is to analyze the alias effect. For convenience, $\mathcal{Q}_\infty(z)$ is defined as the un-saturated quantization function.

2.1 Probability Density Function After Quantization

The nice exposition of quantization seen as area sampling from [23] is reviewed here. Define the probability function as

$$p_i = \text{Prob}\left(y = i\Delta + \frac{\Delta}{2}\right), \quad i = -m, \ldots, m-1, \tag{3}$$

and consider a stochastic signal e with pdf $p_e(e)$. If the measurement is quantized, i.e., $y = \mathcal{Q}_m(e)$, then

$$p_i = \int_{i\Delta}^{(i+1)\Delta} p_e(e)\, de. \tag{4}$$

This integral can equivalently be expressed as convolving the noise distribution $p_e(y)$ with a uniform distribution

$$p_U(y) = \begin{cases} \frac{1}{\Delta}, & -\frac{\Delta}{2} \le y \le \frac{\Delta}{2}, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

followed by sampling in the regular points $i\Delta + \frac{\Delta}{2}$, [23]. Defining the pulse train $l(y) = \sum_{i=-m}^{m-1} \delta\left(y - i\Delta + \frac{\Delta}{2}\right)$, the discrete pdf for y is given as

$$p_y(y) = l(y)(p_e \star p_U)(y) = \sum_{i=-m}^{m-1} \delta\left(y - i\Delta + \frac{\Delta}{2}\right) \int p_e(s) p_U(y-s)\, ds, \tag{6}$$

where '$\star$' denotes the convolution operator.

2.2 Aliasing in the Characteristic Function

The *characteristic function* (CF) defined as the *Fourier transform* (FT) of the pdf is given as

$$\Phi_y(u) = \mathcal{F}\{p(y)\} = \mathbb{E}\left(\mathrm{e}^{juy}\right) = \int_{-\infty}^{\infty} \mathrm{e}^{juy} p(y)\, dy. \tag{7}$$

Note that the frequency axle is reversed compared to the usual definition of the FT, but here the CF will be referred to as the FT of the pdf. Hence, with $L(u) = \mathcal{F}\{l(y)\}$, (6) implies that the CF for y is

$$\Phi_y(u) = L \star (\Phi_e \Phi_u)(u) = \sum_{k=-\infty}^{\infty} \Phi_e\left(u + k\frac{2\pi}{\Delta}\right) \operatorname{sinc}\left(\frac{\Delta(u + k\frac{2\pi}{\Delta})}{2}\right), \quad (8)$$

where $\Phi_u(u) = \operatorname{sinc}(\Delta u/2) = \sin(\Delta u/2)/(\Delta u/2)$. For details on characteristic functions, see for instance [7, 19].

Table 2.2 summarizes the similarities between sampling and quantization as given in [23]. From (8), a kind of quantization 'aliasing' is introduced, similar to Poisson's summation formula. This can be avoided if the CF is 'band limited'. Such an 'anti-alias' condition for quantization is thus

$$\Phi_e(u) = 0, \quad |u| > \pi/\Delta. \quad (9)$$

In the sequel the terms band limited, anti-alias and Poisson's summation formula will be used for both sampling and quantization. Clearly, standard pdfs, as the Gaussian one, do not satisfy band-limitedness. The CF for $e \in \mathcal{N}(0, \sigma^2)$ is

$$\begin{aligned}\Phi_e(u) = \mathbb{E}\left(\mathrm{e}^{juy}\right) &= \int_{-\infty}^{\infty} \mathrm{e}^{juy} \frac{1}{\sqrt{2\pi\sigma}} \mathrm{e}^{-\frac{1}{2\sigma^2}y^2}\, dy = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma}} \mathrm{e}^{-\frac{1}{2\sigma^2}(y^2 - 2juy\sigma^2)}\, dy \\ &= \mathrm{e}^{-\frac{(u\sigma)^2}{2}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma}} \mathrm{e}^{-\frac{1}{2\sigma^2}(y - ju\sigma)^2}\, dy = \mathrm{e}^{-\frac{(u\sigma)^2}{2}}. \quad (10)\end{aligned}$$

Note that one does not obtain a band-limited noise by simply truncating $\Phi_e(u) = \mathrm{e}^{-(u\sigma)^2/2}$ for $|u| > \pi/\Delta$, since then the pdf will lose its positiveness.

That is, standard pdf's imply *quantization aliasing*, which means that high frequencies (fast variations) in the pdf will be interpreted as low frequencies (slow variations).

2.3 Reconstruction of CF and pdf

It follows directly from (8) that the CF for the non-quantized measurement can be reconstructed as

$$\Phi_e(u) = \frac{\Phi_y(u)}{\operatorname{sinc}(\Delta u/2)}, \quad (11)$$

if the anti-alias condition (9) is satisfied, and thus the complete pdf can be constructed.

2.4 Reconstruction of Moments

One useful property of the CF is that all higher order moments can be calculated from it, [23]. This follows from the Taylor expansion $\Phi_e(u) = \mathbb{E}\left(\mathrm{e}^{jue}\right) = 1 +$

Table 1: *Comparison of sampling and un-saturated quantization.*

Feature	Sampling	Quantization
Signal $z(t)$	$z_k = z(kT)$	$y_k = \Delta \left\lfloor \frac{z_k}{\Delta} \right\rfloor + \frac{\Delta}{2}$
Stochastic description	Covariance function $R_z(\tau)$	PDF $p_y(y)$
Fourier characterization	Spectrum $\Phi_z(\omega) = \mathcal{F}\{R_z(\tau)\}$	CF $\Phi_y(u) = \mathcal{F}\{p_y(y)\}$
Poisson's formula	$\Phi_z(\omega) = \sum_{l=-\infty}^{\infty} \Phi_x\left(\omega + l\frac{2\pi}{T}\right)$	$\Phi_y(u) = \sum_{l=-\infty}^{\infty} \Phi_y\left(u + l\frac{2\pi}{\Delta}\right) \operatorname{sinc}\left(\frac{\Delta(u+l\frac{2\pi}{\Delta})}{2}\right)$
Anti-alias condition	$\Phi_x(\omega) = 0, \quad \lvert\omega\rvert > \frac{\pi}{T}$	$\Phi_y(u) = 0, \quad \lvert u\rvert > \frac{\pi}{\Delta}$
Reconstruction	$\Phi_x(\omega) = \begin{cases} \Phi_z(\omega), & \lvert\omega\rvert < \frac{\pi}{T} \\ 0, & \lvert\omega\rvert > \frac{\pi}{T} \end{cases}$	$\Phi_z(u) = \begin{cases} \frac{\Phi_y(u)}{\operatorname{sinc}[\frac{\Delta u}{2}]}, & \lvert u\rvert < \frac{\pi}{\Delta} \\ 0, & \lvert u\rvert > \frac{\pi}{\Delta} \end{cases}$
Moment condition	$\Phi_x(\omega) = 0, \quad \lvert\omega\rvert > \frac{2\pi}{T} - \varepsilon$	$\Phi_y(u) = 0, \quad \lvert u\rvert > \frac{2\pi}{\Delta} - \varepsilon$
Reconstruction of moments	$\int \tau^r R_x(\tau)\, d\tau = \frac{1}{j^r} \frac{d^r}{d\omega^r} \Phi_z(u)\Big\rvert_{\omega=0}$	$\mathbb{E}(x^r) = \frac{1}{j^r} \frac{d^r}{du^r} \Phi_z(u)$

$ju\mathbb{E}(e) - \frac{1}{2!}u^2\mathbb{E}(e^2) + \ldots$. Hence,

$$\mathbb{E}(e^r) = \frac{1}{j^r} \frac{d^r}{du^r} \Phi_e(u)\Big|_{u=0}. \tag{12}$$

Since the CF needs to be correct only close to the origin, aliasing is here tolerated as long as there is no folding at $u = 0$. That is, a less conservative condition for moment reconstruction is that $\Phi_e(u) = 0$ for $|u| > 2\pi/\Delta - \varepsilon$ for some $\varepsilon > 0$, as stated in [23]. In Example 1, the CF is used to compute the first even moments of quantized Gaussian noise.

Example 1 (Moments of $\mathcal{Q}_\infty(\mathbf{e})$) *Consider the case of quantized Gaussian noise. The moment formula* (12)*, using* (10) *in* (8)*, gives for terms corresponding to* $k = -2, \ldots, 2$ *:*

$$\mathbb{E}(y^2) = \frac{\Delta^2}{12} + \sigma^2 - \left(4\sigma^2 + \frac{\Delta^2}{\pi^2}\right) \cdot \mathrm{e}^{-2\pi^2 \frac{\sigma^2}{\Delta^2}} + \left(4\sigma^2 + \frac{\Delta^2}{4\pi^2}\right) \cdot \mathrm{e}^{-8\pi^2 \frac{\sigma^2}{\Delta^2}}, \tag{13}$$

$$\mathbb{E}(y^4) = 3\sigma^4 + \frac{\Delta^4}{80} + \frac{\sigma^2\Delta^2}{2} + \left(-\frac{\Delta^4}{2\pi^2} - 2\sigma^2\Delta^2 + \frac{6\Delta^2\sigma^2}{\pi^2} + \frac{3\Delta^4}{\pi^4} + \frac{32\sigma^6\pi^2}{\Delta^2}\right) \cdot \mathrm{e}^{-2\pi^2 \frac{\sigma^2}{\Delta^2}}. \tag{14}$$

The first line in each equation describe the AUN effect ($k = 0$), while the second line corresponds to terms due to aliasing ($k = \pm 1$). Figure 2 illustrates the dependence of Gaussian noise standard deviation σ for the case of $\Delta = 1$. As can be seen from (13)*, the alias term is negligible when $\Delta \ll \sigma$, and the critical region is when $\Delta \approx 3\sigma$. Another interesting thing to note is that the AUN and alias contributions almost cancel out when $\Delta \gg 3\sigma$ (note that $1/12 \approx 1/\pi^2 - 1/(4\pi^2)$). In Section 5, the moment-based parameter estimation is discussed in more detail.*

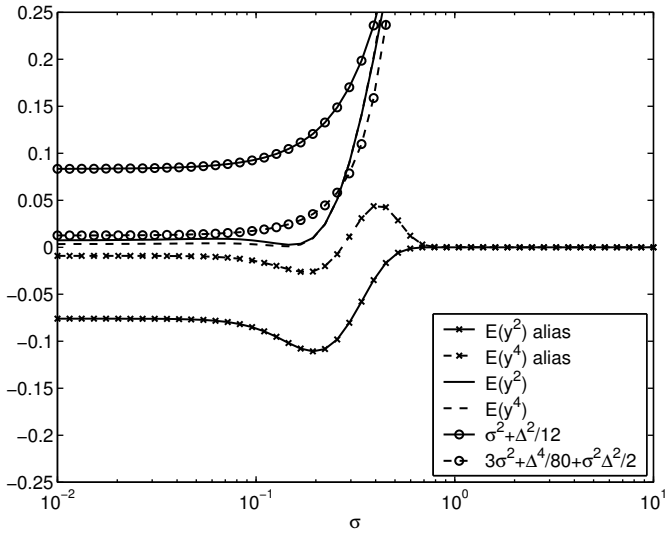


Figure 2: *First non-vanishing moments for quantized Gaussian noise, including the alias effect for different σ using $\Delta = 1$.*

3 Band-Limited Noise and Anti-Alias Noise

The theory in [23] is here extended with some useful results further exploiting the relations to sampling theory. In the sequel, only band-limited noise and aliasing in the quantization sense is discussed.

First, practical methods to obtain band-limited pdf's are given.

3.1 Band-Limited Noise

The class of band-limited signals plays an important role in signal processing because of the possibility of perfect reconstruction of the continuous time signal from a sampled signal. Here, the same question for quantized stochastic signals is studied: Which class of 'band-limited' noise distributions can be perfectly reconstructed?

Here the following class of band-limited pdf's is presented.

Theorem 1 (Band-limited noise) *A sufficient condition for a pdf to be band-limited in the sense* $\Phi(u) = \mathcal{F}\{p(e)\} = 0$ *for* $u > \pi/\Delta$ *is that*

$$\Phi(u) = \mathcal{H} \star \mathcal{H}(u) = \int \mathcal{H}(s)\mathcal{H}^*(u-s)\,ds, \tag{15}$$

where the complex valued function $\mathcal{H}(u)$ *satisfies*

$$\mathcal{H}(u) = 0, \quad \text{for } |u| > \frac{\pi}{2\Delta} \text{ and} \tag{16a}$$

$$\int_{-\frac{\pi}{2\Delta}}^{\frac{\pi}{2\Delta}} |\mathcal{H}(u)|^2\,du = 2\pi. \tag{16b}$$

Proof First, the support of $\mathcal{H}\star\mathcal{H}(u)$ is $[-\pi/\Delta, \pi/\Delta]$, so $\Phi(u)$ is band-limited. Using the well-known Fourier relations $\mathcal{H}\star\mathcal{H}(u) \leftrightarrow |\mathfrak{h}(e)|^2$ and Parseval's formula, (16) immediately gives

$$p(e) = |\mathfrak{h}(e)|^2 > 0, \tag{17a}$$

$$\int p(e)\,de = \int |\mathfrak{h}(e)|^2\,de = \frac{1}{2\pi}\int |\mathcal{H}(u)|^2\,du = 1. \tag{17b}$$

which are the two conditions posed on a pdf. This proves sufficiency. □

3.2 Constructing Band-Limited pdfs

Theorem 1 indicates a constructive method to define noise pdfs that enable perfect reconstruction after quantization. Take an arbitrary pdf $p(e)$. It can be split up as $p(e) = |\mathfrak{h}(e)|^2 = \mathfrak{h}(e)\mathfrak{h}^*(e)$. Denoting its Fourier transform with $\mathcal{H}(u)$, the CF of $p(e)$ can be written $\Phi(u) = \mathcal{H} \star \mathcal{H}(u)$. This problem is quite similar to spectral analysis and the spectral factorization theorem. One major difference here is the phase ambiguity, $\mathcal{H}(u)$ does not need to be minimum phase.

Assume to start with, that $p(e)$ is a symmetric pdf. That is, it is an even function. Hence, $\mathfrak{h}(e)$ can be assumed real, and $\mathcal{H}(u)$ becomes a real function as well. More specifically, the following steps should be performed as presented in Algorithm 1. In the algorithm, it is the truncation step that depends on the

Algorithm 1 Band-limited approximation $\hat{p}(e)$ of a symmetric pdf $p(e)$

1: Define $\mathfrak{h}(e) = \sqrt{p(e)}$.
2: Compute $\mathcal{H}(u) = \mathcal{F}\{\mathfrak{h}(e)\}$.
3: Truncate it to $\hat{\mathcal{H}}(u) = \mathcal{H}(u)$ for $|u| < \frac{\pi}{2\Delta}$.
4: Inverse transform to $\hat{\mathfrak{h}}(e) = \mathcal{F}^{-1}\{\hat{\mathcal{H}}(u)\}$.
5: Compute the normalization constant $c = \left(\int \hat{\mathfrak{h}}^2(e)\,de\right)^{-1}$.
6: Square the result $\hat{p}(e) = c\hat{\mathfrak{h}}^2(e)$.

quantization level and the square guarantees that the pdf is positive.

Example 2 (Gaussian approximation) *Algorithm 1 is here used to approximate the Gaussian standard pdf* $\mathcal{N}(0,1)$ *with a band-limited one. The result for different quantization levels* Δ *and fixed* $\sigma = 1$ *is illustrated in Figure 3. Obviously,* $\Delta = \sigma$ *gives virtually the same pdf.*

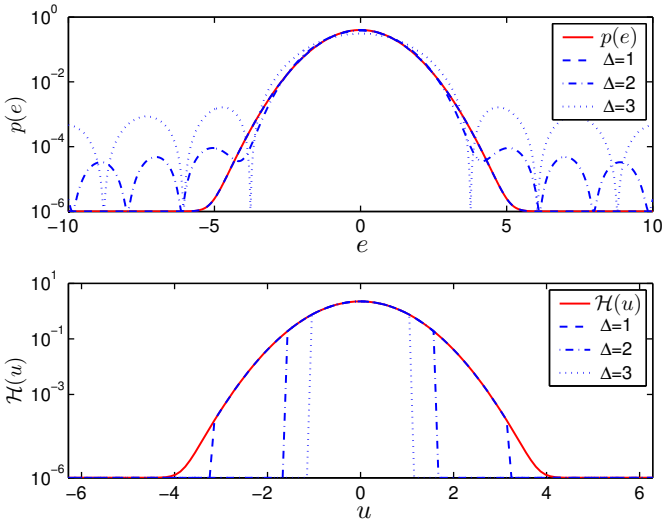


Figure 3: *Band-limited Gaussian approximation (where* 10^{-6} *is added before plotting).*

3.3 Anti-Alias Noise

Unfortunately, perfect anti-alias noise does not exist. The perfect anti-alias noise with a flat CF would have $\mathcal{H} \star \mathcal{H}(u) = c_1$ being constant for $|u| < \pi/\Delta$. That would imply that $p(e) = c_2 \text{sinc}(\pi e/\Delta)$, which is neither positive nor does it integrate to one for any constant c_2.

Take any function $\mathcal{H}(u)$ with compact support. The first try might be a rectangular window, $\mathcal{H}(u) = \sqrt{2\pi}$, so the CF becomes a Bartlett (triangular) window. Then, $p(e) = c_3 \frac{\sin^2(\pi e/\Delta)}{e^2}$ which is positive and integrable. There are two problems with this choice:

- The dithering noise has infinite variance, since $\int e^2 c_3 \frac{\sin^2(\pi e/\Delta)}{e^2}\, de$ does not exist. This can also be seen from the moment formula (12), since the Bartlett window is not differentiable at the origin. This means that moment-based estimators cannot be used.
- The reconstruction formula (23) (see Section 5) includes a division with a function that approaches zeros at the boundaries, so likelihood based methods cannot be used.

One can proceed by taking $\mathcal{H}(u)$ being a Bartlett window, and $p(e) = c_4 \text{sinc}^4(\pi e/\Delta)$. Now, the CF is twice differentiable and the second moment exists. More clever choices should make use of smoother functions $\mathcal{H}(u)$. Here, numerical approaches are possible. One such is the Parks-McClellan's remez algorithm, [15], that returns the real, symmetric FIR filter $\mathfrak{h}(e_i)$ that best approximates the specified $\mathcal{H}(u)$ in a minimax sense.

Any uniform random number generator providing random numbers $v_i \in [0, 1]$ can be used to generate random numbers $e_i = \hat{P}^{-1}(v_i)$ from $\hat{p}(e)$, using its cumulative density function $\hat{P}(e)$.

3.4 Summary

To conclude the section, the correspondences between sampling and quantization are summarized in Table 2, as a complement to Table 2.2.

Table 2: *Further similarities between sampling and quantization.*

Sampling	Quantization
(i). For signal modeling, the class of band-limited signals is of interest.	(i). For sensor modeling, the class of band-limited pdfs is of interest.
(ii). Perfect low-pass filtering with cut-off frequency π/T projects an arbitrary signal to the class of band-limited signals.	(ii). Adding dithering noise with CF $\Phi_d(u)$ having cut-off frequency π/Δ projects an arbitrary sensor noise to the class of band-limited noises.
(iii). The low-pass filter should have flat pass-band in order not to distort the signal, but this filter is not realizable.	(iii). The perfect dithering noise should have flat CF $\Phi_d(u)$, or $\Phi_d(u) = 1/\text{sinc}(\Delta u/2)$, to enable simple reconstruction, but such a noise is not realizable.

4 Moment-based Parameter Estimation

Mean estimation is based on the following signal model:

$$z_t = x + e_t + d_t, \tag{18a}$$

$$y_t = \mathcal{Q}_m(z_t). \tag{18b}$$

Here the signal mean x is the unknown parameter, e_t is the physical sensor noise and d_t denotes an optional user generated dithering noise that can be added before quantization.

The following example illustrates that the sample average can be a poor estimator of x, and that dithering can improve the bias more than its negative effect on the variance, so the total mean square error decreases.

Example 3 (Influence of dithering for the sample mean) *Consider the sample mean* $\hat{x} = \frac{1}{N}\sum_{k=1}^{N} y_k$ *for a quantized signal with* $\Delta = 1$*, where dithering noise is added so* $y_k = \mathcal{Q}_\infty(x + d_k)$*. A Monte Carlo simulation using Gaussian noise gives bias and variance in the mean as depicted in Figure 4. A good compromise between bias and variance is achieved for* $\sigma = 0.3$*, which gives a good over-all* root mean square error *(RMSE).*

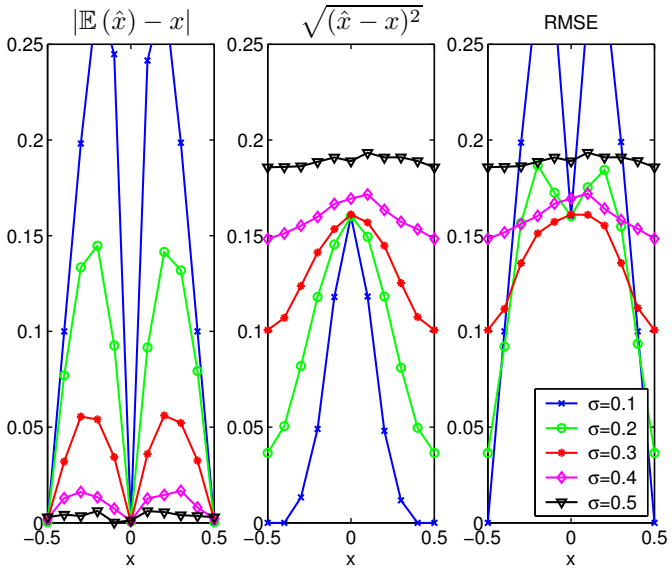


Figure 4: *Bias, standard deviation and RMSE for the sample mean of quantized data for different noise variances in Gaussian dithering noise.*

The AUN approximation is clearly insufficient for understanding the result in this example, since the second order properties of the Gaussian and uniform dithering noise are the same. That is, it is the alias effects that makes the difference. It is quite easy to verify that the CF for Gaussian noise is better concentrated in the frequency domain than for uniform noise. Example 3 used the sample average as estimator. A *moment-based estimator* for x is in general a better alternative. It is

defined as a nonlinear equation system of the form

$$\frac{1}{N}\sum_{t=1}^{N} y_t \approx \mathbb{E}\,(y) = g_1(x, p_e, p_d), \tag{19a}$$

$$\frac{1}{N}\sum_{t=1}^{N} y_t^2 \approx \mathbb{E}\left(y^2\right) = g_2(x, p_e, p_d), \tag{19b}$$

$$\frac{1}{N}\sum_{t=1}^{N} y_t^3 \approx \mathbb{E}\left(y^3\right) = g_3(x, p_e, p_d), \tag{19c}$$

and so on. The system of equations can be used both for estimating x and unknown parameters in the noise distribution, as for instance σ_e^2.

Consider once more Example 1. A moment-based estimator [12] of the Gaussian noise variance can be derived using either (13) or (14). Just solve one of these equation for σ^2, where the left-hand side is the observed corresponding sample moment. Since these expressions take both the AUN and alias effect of quantization into account, the estimate will be unbiased. If there is an unknown mean x in the signal, then similar expressions can be derived. The CF for $z \in \mathcal{N}(x, \sigma^2)$ can be shown to be

$$\Phi_z(u) = \mathrm{e}^{-(u\sigma)^2/2 + jux}. \tag{20}$$

The counterparts of (13) and (14) will then yield the two equations (19a-b) in the two unknowns, x and σ^2. The nonlinear equation system will not look very appealing, mainly because of the many alias terms. However, using dithering noise d_t, the system of equations will be simplified. In this way, the equation system will become more attractive for numerical methods.

The design of dithering noise can be based on numeric simulation results as the example below illustrates. However, it is clear that a deeper understanding on the different contributing effects is needed.

Example 4 (Moments of $\mathcal{Q}_\infty(\mathbf{x} + \mathbf{e})$) *Consider again Example 1, but add a mean x to the noise. The moment formula* (12) *applied to Poisson's summation* (8) *using* (20) *rather than* (10) *gives the theoretical bias and standard deviation according to Figure 5.*

The conclusion is that the moment formula (12), Poisson's summation formula (8) and the CF of the dithering noise provide all information needed to assess performance of any moment based estimator. Furthermore, dithering noise simplifies the numerical computation of the estimate.

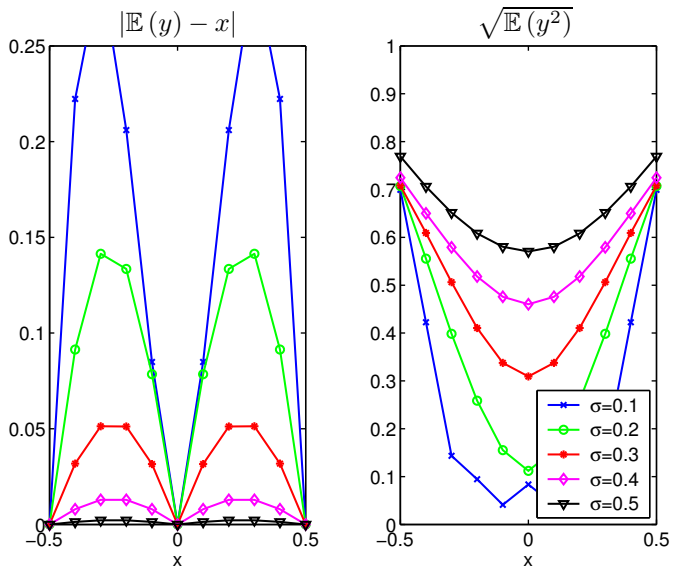


Figure 5: *First non-vanishing moments for $\mathcal{Q}_m(x+e)$ including the alias effect for different σ using $\Delta = 1$.*

5 ML-based Parameter Estimation and Information Bounds

The ML estimation approach is based on maximizing the likelihood $\hat{x} = \arg\max_x p(y_i|x)$. First, it is investigated how the likelihood can be reconstructed using band-limited dithering noise, then Cramér-Rao lower bound and likelihood estimators are derived.

5.1 Reconstruction of Likelihoods

The goal in this section is to get the theoretical relation between the likelihoods

$$p(y_i|x) \leftrightarrow p(z_i|x), \tag{21}$$

where the desired likelihood $p(z_i|x)$ is un-computable, and the computable likelihood $p(y_i|x)$ suffers from quantization effects. Clearly, if the pdf p_e is band-limited and $d_i = 0$, the likelihood can be reconstructed using its CF as

$$\Phi_{z_i|x}(u) = \frac{\Phi_{y_i|x}(u)}{\operatorname{sinc}\left(\frac{\Delta u}{2}\right)}, \quad |u| < \frac{\pi}{\Delta}. \tag{22}$$

This is the basic idea that will be exploited in both likelihood and Bayesian estimation methods in later sections.

Another interesting question is whether there exists a counterpart to an anti-alias filter for ML estimation. It follows from above that any *dithering* noise with pdf satisfying (9) can be added to z before quantization.

Theorem 2 (Anti-alias noise) *Take any band-limited noise d_i (for instance using Theorem 1) and add to the signal $z_i = x + e_i + d_i$ before quantization, so $y_i = \mathcal{Q}_m(x + e_i + d_i)$. Then, the likelihood function can be reconstructed as*

$$\Phi_{z_i|x}(u) = \frac{\Phi_{y_i|x}(u)}{\Phi_d(u)\mathrm{sinc}\left(\frac{\Delta u}{2}\right)}, \quad |u| < \frac{\pi}{\Delta}. \tag{23}$$

Furthermore, the moments $\mathbb{E}(y^r)$ can be expressed as a function of p_e, p_d and Δ without any alias terms if the CF for d_i is band-limited to $2\pi/\Delta - \varepsilon$ for some $\varepsilon > 0$.

Proof The new noise $e+n$ has pdf $p_e \star p_d(z)$ and CF $\Phi_e(u)\Phi_d(u)$ which is band-limited. In the reconstruction (22), the influence of the dithering noise has to be removed. □

That is, the dithering noise d plays the role of an anti-alias filter that is applied before quantization, just as a lowpass filter is applied before sampling.

Adding noise inevitably destroys information. This is in perfect analogy with lowpass filtering used to avoid frequency aliasing. Information is thus lost, but it is at least not mis-interpreted as false information. The same conclusion apply for quantization. Adding a suitably designed dithering noise should increase estimation performance on the quantized data, if proper reconstruction is applied.

The noise design includes choosing a noise that destroys as little information as possible by adapting it to the quantization level (just as choosing the anti-alias lowpass filter as close to the sampling frequency as possible).

5.2 Information Bounds for Parameter Estimation

In the sequel, the analysis is heavily based on expressions involving gradients of scalar functions or vector valued functions. The gradient is defined as:

$$\nabla_x g(x) = \begin{pmatrix} \frac{\partial g}{\partial x_1} \\ \vdots \\ \frac{\partial g}{\partial x_n} \end{pmatrix}, \; g : \mathbb{R}^n \mapsto \mathbb{R}, \tag{24a}$$

$$\nabla_x g^T(x) = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_m}{\partial x_1} \\ \vdots & & \vdots \\ \frac{\partial g_1}{\partial x_n} & \cdots & \frac{\partial g_m}{\partial x_n} \end{pmatrix}, \; g : \mathbb{R}^n \mapsto \mathbb{R}^m. \tag{24b}$$

Also, the Laplacian for the scalar function $g(x, y)$ with $x \in \mathbb{R}^n, y \in \mathbb{R}^m$ is defined as

$$\boldsymbol{\Delta}_y^x g(x, y) = \nabla_y (\nabla_x g(x, y))^T, \; g : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}. \tag{25}$$

For an unbiased estimator, $\mathbb{E}(\hat{x}) = x$, the CRLB, [4, 12, 13], is given by

$$\mathrm{Cov}(x - \hat{x}) = \mathbb{E}\left((x - \hat{x})(x - \hat{x})^T\right) \succeq J^{-1}(x), \tag{26a}$$

$$J(x) = \mathbb{E}\left(-\boldsymbol{\Delta}_x^x \log p(y|x)\right), \tag{26b}$$

where $J(x)$ denotes the *Fisher information matrix* (FIM) in the measurement y regarding the stochastic variable x and $\boldsymbol{\Delta}$ is the Laplacian operator. Also note that an equivalent representation of the information, [12], is

$$\mathbb{E}\left(\nabla_x \log p(y|x)(\nabla_x \log p(y|x))^T\right) = \mathbb{E}\left(-\boldsymbol{\Delta}_x^x \log p(y|x)\right), \tag{27}$$

where ∇_x denotes the gradient with respect to x. Particularly, a Gaussian likelihood $p(y|x)$, with measurement covariance R, gives

$$J(x) = H^T(x) R^{-1} H(x), \tag{28}$$

where

$$H^T(x) = \nabla_x h^T(x). \tag{29}$$

For the case with independent measurements $y^{(i)}, i = 1, \ldots, M$, the information is given as

$$J(x) = \sum_{i=1}^{M} J^{(i)}(x), \tag{30}$$

due to the additivity of information, assuming that $J^{(i)}$ is the information for measurement i.

Consider now the problem of estimating x from the quantized measurements $y = \mathcal{Q}_m(x + e)$. Explicit expressions for the information for Gaussian noise are derived in the sequel. The AUN assumption, which as will be shown, can be quite misleading, is presented in (31).

$$J^{\text{approx}}(x) = \frac{1}{\sigma^2 + \frac{\Delta^2}{12}}. \tag{31}$$

The true information depends on x and includes saturation effects. From now on, saturation effects in the quantization will be taken into account.

First, the sign quantizer is given for its simplicity, and then the general multilevel case is treated.

5.2.1 Sign Quantizer

In this section the Fisher information for the sign quantizer is derived.

Theorem 3 *Consider the sign quantizer*

$$y = \mathcal{Q}_1(x + e) = \mathrm{sign}(x + e), \quad e \in \mathcal{N}(0, \sigma^2). \tag{32}$$

The Fisher information is

$$J_1(x) = \frac{\mathrm{e}^{-\frac{x^2}{\sigma^2}}}{2\pi\sigma^2} \frac{1}{(1-\varrho\left(-x/\sigma\right))\varrho\left(-x/\sigma\right)}, \tag{33}$$

where $\varrho\left(x\right) \triangleq \mathrm{Prob}\left(X < x\right)$ *denotes the Gaussian distribution function.*

Proof See Appendix A. □

Since information is additive for independent observations, the following corollary on estimation performance follows immediately.

Corollary 1 *The* Cramér-Rao lower bound *(CRLB) for the sign quantizer, using* N *independent observations*

$$y_i = \mathcal{Q}_1\left(x + e_i\right) = \mathrm{sign}(x + e_i),\ e_i \in \mathcal{N}(0, \sigma_i^2), \tag{34}$$

$i = 1, \ldots, N$ *is given by*

$$\mathrm{Cov}\left(x - \hat{x}\right) \succeq \left(\sum_{i=1}^{N} \frac{\mathrm{e}^{-\frac{x^2}{\sigma_i^2}}}{2\pi\sigma_i^2} \frac{1}{(1-\varrho\left(-x/\sigma_i\right))\varrho\left(-x/\sigma_i\right)} \right)^{-1}. \tag{35}$$

5.2.2 Multi-Level Quantization

The sign quantizer can be generalized to the multi-level quantization case.

Theorem 4 *Consider the multi-level quantizer.*

$$y = \mathcal{Q}_m\left(x + e\right), \quad e \in \mathcal{N}(0, \sigma^2). \tag{36}$$

The Fisher information is

$$\begin{aligned} J_m(x) =& \frac{\left(-\frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{1}{2}(\frac{-m\Delta-x}{\sigma})^2}\right)^2}{\varrho\left(\frac{-m\Delta-x}{\sigma}\right)} \\ &+ \sum_{j=-m+1}^{m-1} \frac{\left(-\frac{1}{\sqrt{2\pi}\sigma}\left(\mathrm{e}^{-\frac{1}{2}(\frac{(j+1)\Delta-x}{\sigma})^2} - \mathrm{e}^{-\frac{1}{2}(\frac{j\Delta}{\sigma})^2}\right)\right)^2}{\varrho\left(\frac{(j+1)\Delta-x}{\sigma}\right) - \varrho\left(\frac{j\Delta-x}{\sigma}\right)} \\ &+ \frac{\left(\frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{1}{2}(\frac{m\Delta-x}{\sigma})^2}\right)^2}{1-\varrho\left(\frac{m\Delta-x}{\sigma}\right)} \end{aligned} \tag{37}$$

where $\varrho\left(x\right) \triangleq \mathrm{Prob}\left(X < x\right)$ *denotes the Gaussian distribution function.*

Proof See Appendix B. □

Again, since information is additive for independent observations, the following corollary follows.

Corollary 2 *The CRLB for the multi-level quantizer, using N independent observations*

$$y_i = \mathcal{Q}_m\left(x + e_i\right), \quad e_i \in \mathcal{N}(0, \sigma_i^2), \tag{38}$$

$i = 1, \ldots, N$ is given by

$$\mathrm{Cov}\left(x - \hat{x}\right) \succeq J_m^{-1}(x) = \left(\sum_{i=1}^{N} \sum_{j=-m}^{m-1} \frac{\left(\frac{\partial p_j(x)}{\partial x}\right)^2}{p_j(x)} \right)^{-1}, \tag{39}$$

where $p_j(x)$ is given in (65).

5.2.3 Regression Information

Consider next the case of a multi-variable unknown parameter x. A linear regression problem with quantized measurements corresponds to the model

$$y_i = \mathcal{Q}_m\left(H_i x + e_i\right). \tag{40a}$$

The nonlinear least squares problem,

$$y_i = \mathcal{Q}_m\left(h_i(x) + e_i\right), \tag{40b}$$

$$H_i = \frac{d}{dx} h_i(x), \tag{40c}$$

can be treated in parallel. The information in the quantized measurements is given in the following theorem.

Theorem 5 *Consider the regression models* (40) *with $e_i \in \mathcal{N}(0, \sigma_i^2)$, $i = 1, \ldots, N$. Let $s_i = H_i x$ and $s_i = h_i(x)$ denote the signal part for linear and nonlinear regression, respectively. The Fisher information for quantized measurements is given by*

$$J(x) = \sum_{i=1}^{N} H_i^T J_m(s_i) H_i. \tag{41}$$

Proof Follows from the chain rule, the additivity of information and calculations according to Theorem 3 and Theorem 4, respectively. □

5.2.4 Illustration

The following example illustrates how the information and thus the CRLB depends on the quantization level.

Example 5 (CRLB – Multi-level quantizer) *In Figure 6, the Fisher information $J_m(x)$ is illustrated by plotting the lower bound $J_m^{-1/2}(x)$ on the standard deviation for different quantization levels $\Delta = 2/m$. Here, the midriser quantizer with additive noise, $y = \mathcal{Q}_m(x+e)$, $e \in \mathcal{N}(0,\sigma^2)$ is used with $\sigma = 0.1$. Note that $J_1^{-1/2}00(x) \approx \sigma$ and that J_m converges to the AUN in (31) when $m \to \infty$.*

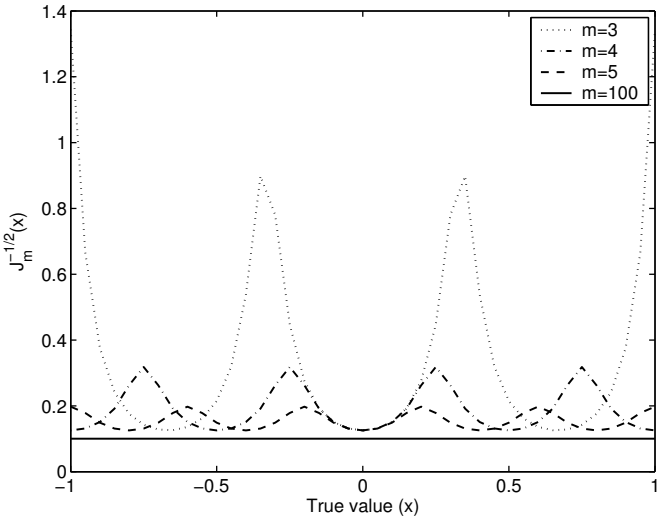


Figure 6: *Fisher information used to compute the standard deviation lower bound $J_m^{-1/2}(x)$ as a function of x for different quantization levels $\Delta = 2/m$.*

5.3 ML-based Estimation

The set of quantized measurements will be denoted $\mathbb{Y}_t = \{y_t^{(i)}\}_{i=1}^N$ and the non-quantized set $\mathbb{Z}_t = \{z_t^{(i)}\}_{i=1}^N$.

5.3.1 ML for Sign Quantization

Form the log-likelihood as

$$\begin{aligned}\log p(\mathbb{Y}|x) =& \log \prod_{i=1}^{N} p(y^{(i)}|x) = \sum_{i=1}^{N} \log p(y^{(i)}|x) = \\ =& N_- \log \varrho\left(-x/\sigma\right) + N_+ \log(1 - \varrho\left(-x/\sigma\right)),\end{aligned} \tag{42}$$

where N_- and N_+ denote the number of terms with $y^{(i)} = -1$ and $y^{(i)} = +1$ respectively, so that $N_- + N_+ = N$. Maximizing the expression by differentiation yields

$$\frac{N_+}{N_-} = \frac{1 - \varrho\left(-x^{\mathrm{ML}}/\sigma\right)}{\varrho\left(-x^{\mathrm{ML}}/\sigma\right)}. \tag{43}$$

Hence

$$\varrho\left(-x^{\mathrm{ML}}/\sigma\right) = \frac{N_-}{N_- + N_+} = \frac{N_-}{N}. \tag{44}$$

Since the left hand side is a monotone and increasing function, the estimate, $\hat{x}^{\mathrm{ML}}$, can be found with a simple line search. For more information on sign quantizers, see for instance [9], where the ML and CRLB for the frequency are calculated for a sinusoidal in noise.

5.3.2 ML for Multi-Level Quantization

For several measurements, the log-likelihood is

$$\log p(\mathbb{Y}_N|x) = \sum_{i=1}^{N} \log p(y^{(i)}|x) = \sum_{j=-m}^{m} N_j \log p_j(x), \tag{45}$$

where N_j is the number of occurrences of each $y^{(j)}$, so that $\sum_j N_j = N$. The ML estimate is here found numerically by searching for maximum of (45). Here $p_j(x)$ is given by (65) for the case $y_i = \mathcal{Q}_m\left(x + e_i\right)$. This is easily extended to linear and nonlinear regression problems (40). In Example 6 the multi-level quantization is presented.

Example 6 (Estimation – multi-level quantizer) *Consider the multi-level quantizer $y = \mathcal{Q}_m\left(x + e\right)$, with $m = 3, \Delta = 0.5$, using the midriser convention. The noise is assumed independent and $e \in \mathcal{N}(0, 0.14^2)$. In Figure 7, the CRLB and the standard deviation for the ML-estimate using 100 Monte Carlo simulations are presented, as a function of the true value x, using $N = 20$ measurements. The approximation, assuming additive noise $y = \mathcal{Q}_m\left(x + e\right) \approx x + e + n, \mathrm{Var}\left(n\right) = \frac{\Delta^2}{12} \approx \mathrm{Var}\left(e\right)$ is also given, where $\sigma^2_{approx} = \frac{1}{N}(\mathrm{Var}\left(e\right) + \mathrm{Var}\left(n\right))$.*

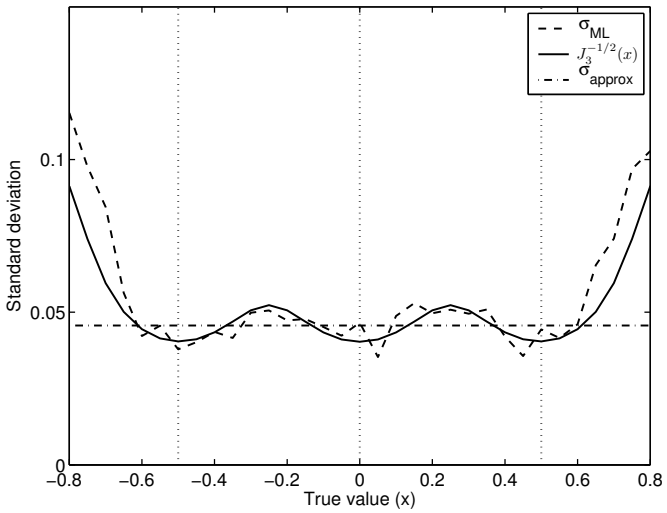


Figure 7: *The CRLB $J_3^{-1/2}(x)$ and ML standard deviation as a function of the true value compared to the additive noise approximation, σ_{approx}, when $m = 3$ levels are used..*

6 State Estimation and Information Bounds

For dynamic systems the following model is considered

$$x_{t+1} = f(x_t, w_t), \tag{46a}$$
$$z_t = h(x_t) + e_t, \tag{46b}$$
$$y_t = \mathcal{Q}_m(z_t). \tag{46c}$$

The Bayesian solution to this estimation problem is given by, [10],

$$p(x_{t+1}|\mathbb{Y}_t) = \int_{\mathbb{R}^n} p(x_{t+1}|x_t) p(x_t|\mathbb{Y}_t)\, dx_t, \tag{47a}$$
$$p(x_t|\mathbb{Y}_t) = \frac{p(y_t|x_t) p(x_t|\mathbb{Y}_{t-1})}{p(y_t|\mathbb{Y}_{t-1})}, \tag{47b}$$

where $p(x_{t+1}|\mathbb{Y}_t)$ is the prediction density and $p(x_t|\mathbb{Y}_t)$ the filtering density. The problem is in general not analytically solvable. There are two fundamentally different ways to approach filtering of nonlinear non-Gaussian dynamic systems:

- The *extended Kalman filter* (EKF), [1, 11], that is the sub-optimal filter for an approximate linear Gaussian model using the AUN assumption, or the optimal *linear* filter for linear non-Gaussian systems.

- Numerical approaches, such as the *particle filter* (PF) [6, 5, 16], that give an arbitrary good approximation of the optimal solution to the Bayesian filtering problem.

These two approaches are compared below.

6.1 Posterior CRLB

The theoretical posterior CRLB for a dynamic system is analyzed in [5, 21, 20, 2]. Here, a quantized sensor using the following model is considered:

$$x_{t+1} = f(x_t, w_t), \tag{48a}$$
$$y_t = \mathcal{Q}_m\left(h(x_t) + e_t\right). \tag{48b}$$

From [2], the posterior CRLB is

$$\operatorname{Cov}\left(x_t - \hat{x}_{t|t}\right) = \mathbb{E}\left((x_t - \hat{x}_{t|t})(x_t - \hat{x}_{t|t})^T\right) \succeq P_{t|t}, \tag{49}$$

where $P_{t|t}$ can be retrieved from the recursion

$$P_{t+1|t+1}^{-1} = Q_t^{-1} + J_{m,t+1} - S_t^T \left(P_{t|t}^{-1} + V_t\right)^{-1} S_t, \tag{50}$$

where

$$V_t = \mathbb{E}\left(-\boldsymbol{\Delta}_{x_t}^{x_t} \log p(x_{t+1}|x_t)\right), \tag{51a}$$
$$S_t = \mathbb{E}\left(-\boldsymbol{\Delta}_{x_t}^{x_{t+1}} \log p(x_{t+1}|x_t)\right), \tag{51b}$$
$$Q_t^{-1} = \mathbb{E}\left(-\boldsymbol{\Delta}_{x_{t+1}}^{x_{t+1}} \log p(x_{t+1}|x_t)\right), \tag{51c}$$
$$J_{m,t} = \mathbb{E}\left(-\boldsymbol{\Delta}_{x_t}^{x_t} \log p(y_t|x_t)\right). \tag{51d}$$

Hence, the measurement quantization effects will only affect $J_{m,t}$, which is given by Theorem 4. For linear dynamics with additive Gaussian noise

$$x_{t+1} = F_t x_t + w_t, \tag{52}$$

the following holds

$$V_t = F_t Q_t^{-1} F_t^T, \quad S_t = -F_t^T Q_t^{-1}, \tag{53}$$

where $\operatorname{Cov}(w_t) = Q_t$.

6.2 Kalman Filter for Measurement Quantization

Consider the following linear Gaussian model with quantized observations:

$$\begin{aligned} x_{t+1} &= F_t x_t + G_t w_t, & \operatorname{Cov}(w_t) &= Q_t, \\ z_t &= H_t x_t + e_t, & \operatorname{Var}(e_t) &= \sigma^2, \\ y_t &= \mathcal{Q}_m(z_t). \end{aligned}$$

The quantized measurement, y_t, is treated as a scalar, but the multi-variable case is covered as long as the measurement noises $e_{t,i}$ are independent, using measurement update iterations in the Kalman filter. Using the AUN assumption, the optimal filter is given by the Kalman filter by increasing the measurement covariance by $\frac{\Delta^2}{12}$, i.e.,

$$R_t = \sigma_t^2 + \frac{\Delta^2}{12}\mathcal{I}, \tag{54}$$

where $\mathcal{I}$ is the identity matrix. Remember that the AUN assumption does not include saturation nor correlation properties from the quantization. In [24], the finite word-length for Kalman filter implementation is discussed in more detail.

6.3 Particle Filter for Measurement Quantization

The particle filter, [6, 5, 16], here adopted to quantized measurements is given in Algorithm 2. Note that quantization is treated formally correct by using its theoretical likelihood in (55).

Algorithm 2 The particle filter.

1: Set $t = 0$. For $i = 1, \ldots, N_{\mathrm{PF}}$, initialize the particles, $x_{0|-1}^{(i)} \sim p_{x_0}(x_0)$.

2: For $i = 1, \ldots, N_{\mathrm{PF}}$, evaluate the importance weights $\gamma_t^{(i)} = p(y_t|x_t^{(i)})$ according to the likelihood

$$p(y_t|x_t) = p_j(x_t), \tag{55}$$

where $p_j(x)$ is given in Appendix B.

3: Resample N_{PF} particles with replacement according to,

$$\mathrm{Prob}(x_{t|t}^{(i)} = x_{t|t-1}^{(j)}) = \tilde{\gamma}_t^{(j)},$$

where the normalized weights are given by

$$\tilde{\gamma}_t^{(i)} = \frac{\gamma_t^{(i)}}{\sum_{j=1}^{N_{\mathrm{PF}}} \gamma_t^{(j)}}.$$

4: For $i = 1, \ldots, N_{\mathrm{PF}}$, predict new particles according to

$$x_{t+1|t}^{(i)} \sim p(x_{t+1|t}|x_t^{(i)}).$$

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

For hardware implementations, for instance on efficient resampling algorithms and on the complexity and performance issue for quantized particle filters, see [8, 3]. In [17, 18], the particle filter method is proposed for a sensor fusion method involving quantization.

6.4 Illustration

In the following example, the sign quantizer for a dynamic system is illustrated.

Example 7 (Filtering – sign quantizer) *Consider the following scalar system with a sign quantizer*

$$\begin{aligned} x_{t+1} &= F_t x_t + w_t, \qquad x_0 = 0, \\ z_t &= x_t + e_t, \\ y_t &= \mathcal{Q}_1\left(z_t\right), \end{aligned}$$

where

$$F_t = 0.95, Q_t = \mathrm{Var}\left(w_t\right) = 0.10^2, R_t = \mathrm{Var}\left(e_t\right) = 0.58^2.$$

In Figure 8, the root mean square error *(RMSE) for the Kalman and the particle filter are presented using* 200 *Monte Carlo simulations. The measurement noise in the Kalman filter was adjusted in the filter as described in* (54)*. The particle filter used the correct sign quantized likelihood according to* (58) *and* 1000 *particles. The theoretical CRLB is also given in Figure 8, as the solution to* (50)*, which for a general case can be solved using a discrete algebraic Riccati solver. For the scalar case in this example, the covariance (*P*) can be derived analytically as the solution to*

$$P^2 + (QJ + 1 - F^2)/(JF^2)P - Q/(JF^2) = 0,$$

where $J = J_{1,t} = \frac{2}{\pi\sigma^2}$ *is given from* (33)*, using* $x = 0$*.*

7 Conclusions

The implication of quantization on Bayesian, likelihood based or moment-based approaches to estimation and filtering has been studied. For all these approaches, a deep understanding is required of how quantization change the statistics of data used in the estimator, in particular when the quantization level is large compared to the standard deviation of the measurement noise. It is well-known that quantization implies a kind of aliasing effect. It is explained how adding dithering noise can make moment and likelihood reconstruction easier. An open question is what and how much can be gained by dithering and how an optimal dithering noise should be designed.

Further, a detailed study on the Cramér-Rao lower bound was given. Several theoretical results and examples were presented to show that estimators utilizing knowledge of the quantization are superior to conventional estimators, where only the second order properties of the quantization is incorporated. Finally, a dedicated particle filter was given that applies to arbitrary filtering problems, where independent quantized measurements are given.

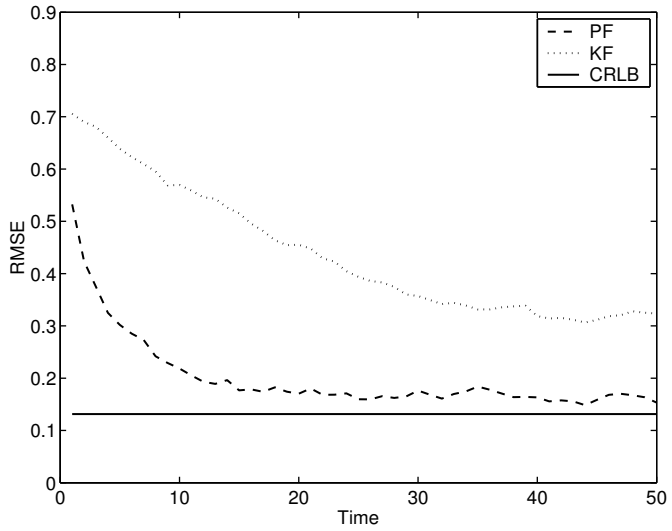


Figure 8: *RMSE for the PF and KF for a linear Gaussian system with a sign quantizer in the measurement relation, compared with the CRLB limit.*

Acknowledgment

This work was supported by the VINNOVA's Center of Excellence ISIS (Information Systems for Industrial Control and Supervision) at Linköping University, Sweden. In particular the partner NIRA Dynamics.

Appendix

A Proof of Theorem 3

Proof The probability function for y can be calculated using

$$\begin{aligned} p(y=-1|x) &= \text{Prob}(x+e<0) = \text{Prob}(e<-x) \\ &= \int_{-\infty}^{-x} \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{t^2}{2\sigma^2}}\, dt = \int_{-\infty}^{-x/\sigma} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{t^2}{2}}\, dt \triangleq \varrho\left(-x/\sigma\right). \end{aligned} \tag{56}$$

Similarly,

$$p(y=+1|x) = \text{Prob}(x+e\geq 0) = 1-\varrho\left(-x/\sigma\right). \tag{57}$$

Hence, the discrete likelihood can be written as

$$p(y|x) = \varrho\left(-x/\sigma\right)\delta(y+1) + (1-\varrho\left(-x/\sigma\right))\delta(y-1), \tag{58}$$

where

$$\delta(i) = \begin{cases} 1, & i = 0, \\ 0, & i \neq 0. \end{cases} \tag{59}$$

To calculate the CRLB variance, apply (26b).

$$J(x) = -\mathbb{E}\left(\frac{\partial^2}{\partial x^2}\log p(y|x)\right) = -\mathbb{E}\left(\frac{\frac{\partial^2 p(y|x)}{\partial x^2}p(y|x) - \left(\frac{\partial p(y|x)}{\partial x}\right)^2}{p^2(y|x)}\right)$$

$$= -\sum_{j\in\{-1,1\}} \frac{\partial^2 p(y=j|x)}{\partial x^2} - \frac{\left(\frac{\partial p(y=j|x)}{\partial x}\right)^2}{p(y=j|x)}. \tag{60}$$

Note that (56) yields

$$\frac{\partial \varrho\left(-x/\sigma\right)}{\partial x} = -\frac{\mathrm{e}^{-\frac{x^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}. \tag{61}$$

Hence

$$\frac{\partial p(y|x)}{\partial x} = \frac{\mathrm{e}^{-\frac{x^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} \times \begin{cases} 1, & y = -1 \\ -1, & y = 1 \end{cases} \tag{62}$$

$$\frac{\partial^2 p(y|x)}{\partial x^2} = \frac{-x\mathrm{e}^{-\frac{x^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma^3} \times \begin{cases} 1, & y = -1 \\ -1, & y = 1 \end{cases} \tag{63}$$

Inserting these equations into (60) gives

$$J(x) = \frac{\mathrm{e}^{-\frac{x^2}{\sigma^2}}}{2\pi\sigma^2} \underbrace{\left(\frac{1}{\left(1-\varrho\left(-x/\sigma\right)\right)} + \frac{1}{\varrho\left(-x/\sigma\right)}\right)}_{\frac{1}{(1-\varrho(-x/\sigma))\varrho(-x/\sigma)}}, \tag{64}$$

which proves the theorem. □

B Proof of Theorem 4

Proof Calculate the probability for each level $j = -m+1, \ldots, m-1$ (see Figure 1) as

$$p_j(x) \triangleq \text{Prob}\left(y = j\Delta + \frac{\Delta}{2}\right) = \text{Prob}\left(j\Delta < x + e \leq (j+1)\Delta\right)$$

$$= \varrho\left(\frac{(j+1)\Delta - x}{\sigma}\right) - \varrho\left(\frac{j\Delta - x}{\sigma}\right). \tag{65a}$$

The probability at the end points are calculated as

$$p_{-m}(x) = \varrho\left(\frac{-m\Delta - x}{\sigma}\right), \tag{65b}$$

$$p_{m-1}(x) = 1 - \varrho\left(\frac{m\Delta - x}{\sigma}\right). \tag{65c}$$

Similar to the sign quantizer, the likelihood is given as

$$p(y|x) = \sum_{j=-m}^{m} p_j(x)\delta\left(y - j\Delta - \frac{\Delta}{2}\right). \tag{66}$$

Proceeding in the same way as for the sign quantizer, note that

$$\frac{\partial p(y|x)}{\partial x} = \sum_{j=-m}^{m} \frac{\partial p_j(x)}{\partial x}\delta\left(y - j\Delta - \frac{\Delta}{2}\right), \tag{67}$$

$$\frac{\partial^2 p(y|x)}{\partial x^2} = \sum_{j=-m}^{m} \frac{\partial^2 p_j(x)}{\partial x^2}\delta\left(y - j\Delta - \frac{\Delta}{2}\right), \tag{68}$$

where the derivatives for $j = -m+1, \ldots, m-2$ are given by

$$\frac{\partial p_j(x)}{\partial x} = -\frac{1}{\sqrt{2\pi}\sigma}\left(\mathrm{e}^{-\frac{1}{2}(\frac{(j+1)\Delta - x}{\sigma})^2} - \mathrm{e}^{-\frac{1}{2}(\frac{j\Delta - x}{\sigma})^2}\right), \tag{69}$$

$$\frac{\partial^2 p_j}{\partial x^2} = -\frac{(j+1)\Delta - x}{\sqrt{2\pi}\sigma^3}\mathrm{e}^{-\frac{1}{2}(\frac{(j+1)\Delta - x}{\sigma})^2} + \frac{j\Delta - x}{\sqrt{2\pi}\sigma^3}\mathrm{e}^{-\frac{1}{2}(\frac{j\Delta - x}{\sigma})^2}. \tag{70}$$

For $j = -m$ or $j = m-1$, differentiating (65) yields

$$\frac{\partial p_{-m}(x)}{\partial x} = -\frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{1}{2}(\frac{-m\Delta - x}{\sigma})^2}, \tag{71a}$$

$$\frac{\partial p_{m-1}(x)}{\partial x} = \frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{1}{2}(\frac{m\Delta - x}{\sigma})^2}, \tag{71b}$$

$$\frac{\partial^2 p_{-m}(x)}{\partial x^2} = -\frac{-m\Delta - x}{\sqrt{2\pi}\sigma^3}\mathrm{e}^{-\frac{1}{2}(\frac{-m\Delta - x}{\sigma})^2}, \tag{71c}$$

$$\frac{\partial^2 p_{m-1}(x)}{\partial x^2} = \frac{m\Delta - x}{\sqrt{2\pi}\sigma^3}\mathrm{e}^{-\frac{1}{2}(\frac{m\Delta - x}{\sigma})^2}. \tag{71d}$$

Note that the terms in (70) form a telescope sum, so together with (71c) and (71d) it yields

$$\sum_{j=-m}^{m-1} \frac{\partial^2 p_j}{\partial x^2} = 0. \tag{72}$$

Hence, the Fisher information is

$$J(x) = \sum_{j=-m}^{m-1}\left(-\frac{\partial^2 p_j}{\partial x^2} + \frac{\left(\frac{\partial p_j(x)}{\partial x}\right)^2}{p_j}\right) = \sum_{j=-m}^{m-1}\frac{\left(\frac{\partial p_j(x)}{\partial x}\right)^2}{p_j}. \tag{73}$$

This, together with (65), (69), (71a) and (71b), proves the theorem. □

References

[1] B.D.O. Anderson and J. B. Moore. *Optimal Filtering.* Prentice Hall, Englewood Cliffs, NJ, 1979.

[2] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Linköping Studies in Science and Technology. Dissertations No. 579, Linköping University, Linköping, Sweden, 1999.

[3] M. Bolic, S. Hong, and P. Djurić. Performance and complexity analysis of adaptive particle filtering for tracking applications. In *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, November 2002.

[4] H. Cramér. *Mathematical Methods of Statistics*. Princeton, NJ: Princeton University Press, 1946.

[5] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[6] N. J. Gordon, D. J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[7] A. Gut. *An Intermediate Course in Probability*. Springer-Verlag, 1995.

[8] S. Hong, M. Bolic, and P. Djurić. An efficient fixed-point implementation of residual resampling scheme for high-speed particle filters. *IEEE Signal Processing Letters*, 11(5):482–485, 2004.

[9] A. Host-Madsen and P. Händel. Effects of sampling and quantization on single-tone frequency estimation. *IEEE Transactions on Signal Processing*, 48(3):650–662, 2000.

[10] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, 1970.

[11] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Information and System Sciences. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[12] S. Kay. *Fundamentals of Statistical Signal Processing*. Prentice Hall, 1993.

[13] E. L. Lehmann. *Theory of Point Estimation*. John Wiley and Sons, 1983.

[14] S. P. Lipshitz, R. A. Wannamker, and J. Vanderkooy. Quantization and dither: A theoretical survey. *Journal of Audio Eng. Soc*, 40(5):355–375, May 1992.

[15] A. Oppenheim and R. Schafer. *Digital Signal Processing*. Prentice-Hall, 1975.

[16] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[17] Y. Ruan, P. Willett, and A. Marrs. Fusion of quantized measurements via particle filtering. In *IEEE Proceedings of Aerospace Conference*, volume 4, pages 1967–1978, March 2003.

[18] Y. Ruan, P. Willett, and A. Marrs. Practical fusion of quantized measurements via particle filtering. In *IEE Target Tracking: Algorithms and Applications,*, pages 13–18, March 2004.

[19] A. Stuart and J. K. Ord. *Kendall's Advanced Theory of Statistics*, volume 1. London: Edward Arnold, New-York Wiley, 6 edition, 1994.

[20] P. Tichavsky, P. Muravchik, and A. Nehorai. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing*, 46(5):1386–1396, 1998.

[21] H. L. Van Trees. *Detection, Estimation and Modulation Theory*. Wiley, New York, 1968.

[22] B. Widrow. A study of rough amplitude quantization by means of Nyquist sampling theory. *IRE Transactions on Circuit Theory*, pages 266–276, December 1956.

[23] B. Widrow, I. Kollar, and M-C Liu. Statistical theory of quantization. *IEEE Transactions on Instrumentation and Measurement*, pages 353–361, April 1996.

[24] D. Williamson. Finite wordlength design of digital Kalman filters for state estimation. *IEEE Transactions on Automatic Control*, pages 930–939, October 1985.

Paper B

Complexity Analysis of the Marginalized Particle Filter

Edited version of the paper:

R. Karlsson, T. Schön, and F. Gustafsson. Complexity analysis of the marginalized particle filter. *To appear in IEEE Transactions on Signal Processing.*

Preliminary version published as Technical Report LiTH-ISY-R-2611, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Complexity Analysis of the Marginalized Particle Filter

Rickard Karlsson, Thomas Schön and Fredrik Gustafsson,

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

Abstract

In this paper the computational complexity of the marginalized particle filter is analyzed and a general method to perform this analysis is given. The key is the introduction of the *equivalent flop* measure. In an extensive Monte Carlo simulation different computational aspects are studied and compared with the derived theoretical results.

Keywords: Nonlinear estimation, Marginalized particle filter, Kalman filter, Complexity analysis, Equivalent Flop.

1 Introduction

In *particle filter* (PF) applications, knowledge of the computational complexity is often of paramount importance. In this paper the computational complexity issues that arise in the use of the *marginalized particle filter* (MPF), also called the Rao-Blackwellized particle filter are studied. The MPF is a clever combination of the standard PF, [10], and the *Kalman filter* (KF), [12], which can be used when the model contains a linear substructure, subject to Gaussian noise. It is a well known fact that in some cases it is possible to obtain better estimates, i.e., estimates with reduced variance, using the MPF instead of using the standard PF [8]. By now quite a lot has been written about the MPF, see e.g., [7, 6, 5, 1, 2, 15]. However, to the best of the authors knowledge, nothing has yet been written about complexity issues. In this article, expressions for the complexity, $\mathcal{C}(p, k, N)$, are derived, where p, k represent the states estimated using the PF and the KF, respectively and N denotes the number of particles. A general method to analyze the computational complexity of the MPF will be provided. The method is illustrated using a common tracking model, but can be applied to a much broader class of models. For more details of the proposed method, the reader is referred to [13].

2 The Marginalized Particle Filter

Many nonlinear estimation problems can be handled using the particle filter. A general state-space model

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

has both nonlinear dynamics, f, and nonlinear measurements, h. The noise processes w_t and e_t have known probability density functions. If the state-space model contains a linear-Gaussian substructure, this can be exploited to obtain better estimates using the MPF. In this article the case with linear-Gaussian dynamics,

$$x_{t+1} = A_t x_t + w_t, \qquad w_t \in \mathcal{N}(0, Q_t), \tag{2a}$$
$$y_t = h(x_t^n) + C_t x_t^l + e_t, \tag{2b}$$

is discussed. In this context the state variable $x_t \in \mathbb{R}^m$ is

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

where $x_t^l \in \mathbb{R}^l$ denote the linear states and $x_t^n \in \mathbb{R}^n$ denotes the nonlinear states. Furthermore, $\mathbb{X}_t^n = \{x_i^n\}_{i=0}^t$ and $\mathbb{Y}_t = \{y_i\}_{i=0}^t$. Using Bayes' theorem,

$$p(\mathbb{X}_t^n, x_t^l | \mathbb{Y}_t) = p(x_t^l | \mathbb{X}_t^n, \mathbb{Y}_t) p(\mathbb{X}_t^n | \mathbb{Y}_t), \tag{4}$$

where $p(\mathbb{X}_t^n|\mathbb{Y}_t)$ is given by the PF and $x_t^l|\mathbb{X}_t^n$ is linear-Gaussian, i.e., $p(x_t^l|\mathbb{X}_t^n, \mathbb{Y}_t)$ is given by the KF. This marginalization idea is certainly not new [7, 4, 8, 5, 1, 8, 15, 14]. The state vector x_t can be partitioned into two parts, $x_t^p \in \mathbb{R}^p$ and $x_t^k \in \mathbb{R}^k$, which are estimated using the PF and the KF respectively. Furthermore, $p \in [n, n+l]$, $k \in [0, l]$ and for the general partitioning case $p - n$ states can be selected from l possibilities.

It is interesting to consider which states to put in the nonlinear and the linear partition, respectively. Two relevant aspects with respect to this partitioning are how it will affect the computational complexity and the estimation performance. This will be discussed using the following model

$$x_{t+1}^p = A_t^p x_t^p + A_t^k x_t^k + w_t^p, \qquad w_t^p \sim \mathcal{N}(0, Q_t^p), \tag{5a}$$
$$x_{t+1}^k = F_t^p x_t^p + F_t^k x_t^k + w_t^k, \qquad w_t^k \sim \mathcal{N}(0, Q_t^k), \tag{5b}$$
$$y_t = h_t(x_t^p) + C_t x_t^k + e_t, \qquad e_t \sim \mathcal{N}(0, R_t), \tag{5c}$$

where the noise is assumed to be independent. This is no restriction, since the case of dependent noise can be reduced to the case of independent noise using a Gram-Schmidt procedure [11]. In Algorithm 1 the MPF is summarized for the model given in (5) (with $C_t = 0$, for the sake of brevity). For a detailed derivation (including the case $C_t \neq 0$), the reader is referred to [15].

3 Complexity Analysis

In this section the computational complexity of the MPF is discussed from a theoretical point of view, by giving the number of *floating-point operations* (flops) used in the algorithm. A flop is here defined as one addition, subtraction, multiplication, or division of two floating-point numbers. However, problems occur when the flop count is compared to the actual computation time. This is due to the fact that issues such as cache boundaries and locality of reference will significantly influence the computation time [3]. Moreover, there are certain steps in the algorithm that cannot easily be measured in flops, for instance the cost of generating a random number and the cost of evaluating a nonlinear function. Despite these drawbacks it is still possible to analyze the complexity using the computer to measure the absolute time that the different steps require. These can then be compared to the theoretical result obtained from counting flops. In the PF the computational complexity of the resampling step is proportional to the number of particles and the amount of time for generating random numbers is proportional to the number of random numbers required. The proportionality coefficients are related to reflect the flop complexity instead of the time complexity for ease of comparison with parts that only depend on matrix and vector operations. This will be referred to as the *equivalent flop* (EF) complexity.

Definition 1 *The* equivalent flop *(EF) complexity for an operation is defined as the number of flops that results in the same computational time as the operation.*

3.1 Nonlinear Measurements ($C_t = 0$)

In this section the case $C_t = 0$ in (5c) is discussed. The total complexity of Algorithm 1 is given for each code line in Table 1. For instance, the first instruction $P_{t|t}(A_t^k)^T$ corresponds to multiplying $P_{t|t} \in \mathbb{R}^{k\times k}$ with $(A_t^k)^T \in \mathbb{R}^{k\times p}$, which requires pk^2 multiplications and $(k-1)kp$ additions [9]. The total EF complexity is given by:

$$\begin{aligned}\mathcal{C}(p,k,N) = 4pk^2 + 8kp^2 + \frac{4}{3}p^3 + 5k^3 - 5kp + 2p^2 + \\ (6kp + 4p^2 + 2k^2 + p - k + pc_3 + c_1 + c_2)N. \end{aligned} \tag{10}$$

Above, the coefficient c_1 has been used for the calculation of the Gaussian likelihood, c_2 for the resampling and c_3 for the random number complexity. Note that, when $C_t = 0$ the same covariance matrix is used for all Kalman filters, which reduces the computational complexity.

Table 1: *The EF complexity for the PF (upper) and KF time update (lower) in Algorithm 1 († represents the case $k > 0$, ‡ represent operations not from matrix multiplications and additions, such as resampling, random number generation etc.).*

Instruction	**Mult.**	**Add.**	**Other**‡
$P_A := P_{t\|t}(A_t^k)^T$	pk^2	$(k-1)kp$	
$M := A_t^k P_A + Q_t^p$	kp^2	$(k-1)p^2 + p^2$ †	
$T_1 := chol(M)$			$\frac{p^3}{3} + 2p^2$
$T_2 := randn(p, N)$			pNc_3
$w := T_1 * T_2$	p^2N	$(p-1)pN$	
$T_3 := A^p x^p$	p^2N	$(p-1)pN$	
$T_4 := A^k x^k$	pkN	$(k-1)pN$ †	
$\hat{x}_{t+1\|t}^p := T_3 + T_4 + w$		$2pN$	
$inv_M := M^{-1}$			p^3
$L := F_t^k P_A inv_M$	$k^2p + kp^2$	$k^2p + p^2k - 2kp$	
$T_5 := F_t^k P_{t\|t}(F_t^k)^T$	$2k^3$	$2(k-1)k^2$	
$T_6 := L_t M_t L_t^T$	$2kp^2$	$2(p-1)pk$	
$P := T_5 + Q_t^k - T_6$		$2k^2$	
$T_7 := F^k x^k$	k^2N	$(k-1)kN$	
$T_8 := F^p x^p$	kpN	$(p-1)kN$	
$T_9 := \hat{x}_{t+1\|t}^p - T_3 - T_4$		$2pN$	
$\hat{x}_{t+1\|t}^k := T_7 + T_8 + LT_9$	kpN	$(p+1)kN$	

The analysis provided above is general and the main steps, which will be discussed in the subsequent section are as follows:

1. Estimate the time for one flop using linear regression.
2. Estimate the time for likelihood calculation, resampling and random number generation.
3. Relate all times using the EF measure.
4. Calculate the overall complexity $\mathcal{C}(p, k, N)$.

By requiring $\mathcal{C}(p+k, 0, N_{\text{PF}}) = \mathcal{C}(p, k, N(k))$, where N_{PF} corresponds to the number of particles used in the standard PF $N(k)$ can be solved for. This gives the number of particles, $N(k)$, that can be used in the MPF in order to obtain the same computational complexity as if the standard particle filter had been used for all states. In Figure 1 the ratio $N(k)/N_{\text{PF}}$ is plotted for systems with $m = 3, \ldots, 9$ states. Hence, using Figure 1 it is possible to directly find out how much there is to gain in using the MPF from a computational complexity point of view. The figure

also shows that the computational complexity is always reduced when the MPF can be used instead of the standard PF. Furthermore, it is well-known that the quality of the estimates will improve or remain the same when the MPF is used [8].

3.2 Mixed Nonlinear/Linear Measurements ($C_t \neq 0$)

It is now assumed that $C_t \neq 0$ in (5c), which implies that the Riccati recursions have to be evaluated for each particle. This results in a significant increase in the computational complexity. Hence, different covariance matrices have to be used for each Kalman filter, implying that (10) has to be modified. For details the reader is referred to [13], but approximately the complexity is given by

$$\begin{aligned}\mathcal{C}(p,k,N) = (&6kp + 4p^2 + 2k^2 + p - k + pc_3 + c_1 + c_2 + \\ &4pk^2 + 8kp^2 + \frac{4}{3}p^3 + 5k^3 - 5kp + 2p^2 + k^3)N.\end{aligned} \tag{11}$$

The problem with the increased complexity in (11) might be reduced simply by moving one or more linear states from x_t^k to x_t^p. In Figure 2 the ratio $N(k)/N_{PF}$ is plotted for systems with $m = 3, \ldots, 9$ states. For systems with few states the MPF is more efficient than the standard PF. However, for systems with more states,

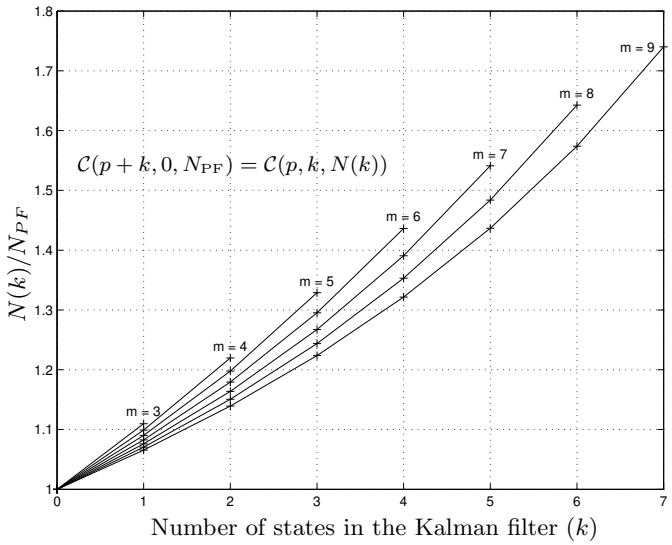


Figure 1: *The ratio $N(k)/N_{\mathrm{PF}}$ for systems with $m = 3, \ldots, 9$ states and $C_t = 0$, $n = 2$ is shown. It is apparent the MPF can use more particles for a given computational complexity, when compared to the standard PF.*

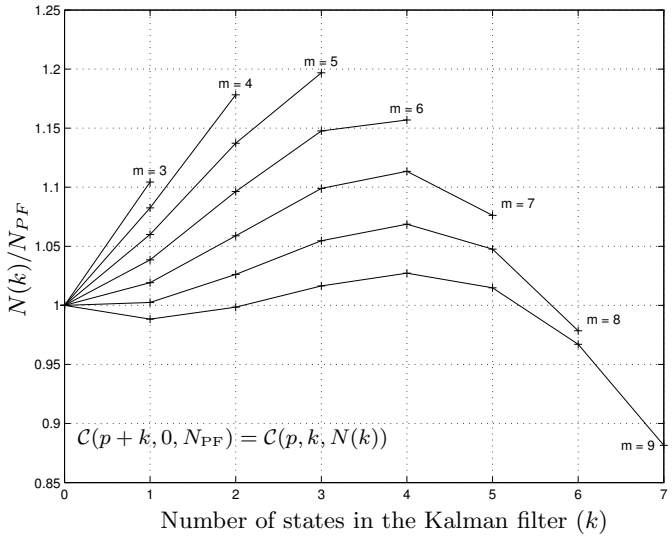


Figure 2: *The ratio $N(k)/N_{PF}$ for systems with $m = 3, \ldots, 9$ states and $C_t \neq 0$, $n = 2$ is shown. For systems with high state dimension and many marginalized states the standard PF can use more particles than the MPF.*

where most of the states are marginalized the standard PF becomes more efficient than the MPF. The reason is the increased complexity in the Kalman filters due to the increased dimension in the Riccati recursions. For example; according to Figure 2 a system with 9 states, where 7 are marginalized, $N(k) < N_{PF}$.

4 Target Tracking Example

The general method for analyzing the computational complexity presented in the previous section is illustrated using a common tracking model. The problem of

estimating the position and velocity of an aircraft is studied using

$$x_{t+1} = \begin{pmatrix} 1 & 0 & T & 0 & T^2/2 & 0 \\ 0 & 1 & 0 & T & 0 & T^2/2 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} x_t + w_t, \tag{12a}$$

$$y_t = \begin{pmatrix} \sqrt{p_x^2 + p_y^2} \\ \arctan\left(p_y/p_x\right) \end{pmatrix} + e_t \tag{12b}$$

where $Q = \mathrm{Cov}\,(w) = \mathrm{diag}(1\ 1\ 1\ 1\ 0.01\ 0.01), R = \mathrm{Cov}\,(e) = \mathrm{diag}(100\ 10^{-6})$ and the state vector is $x_t = (p_x\ p_y\ v_x\ v_y\ a_x\ a_y)^T$, i.e., position, velocity and acceleration. The measurement equation gives the range and azimuth from the radar system.

In the subsequent section a numerical study of the computational complexity is given, where the theoretical expressions previously derived are validated. Furthermore the MPF will be analyzed in an extensive *Monte Carlo* (MC) simulation using the model described in (12). The main purpose of this simulation is to illustrate the implications of the results derived in this paper. In the simulations one state trajectory with different noise realizations have been used. The purpose of the simulations presented here is to show that using marginalization the computational complexity is significantly reduced and the quality of the estimates is improved.

4.1 Numerical Complexity Analysis

The model (12) has 2 nonlinear state variables and 4 linear state variables, implying $k \in [0, 4],\ p \in [2, 6]$. Two cases are now studied, the full PF, where all states are estimated using the PF and the completely marginalized PF, where all linear states are marginalized out and estimated using the KF. Requiring the same computational complexity, i.e., $\mathcal{C}(6, 0, N_{PF}) = \mathcal{C}(2, 4, N_{MPF})$, gives

$$N_{\mathrm{PF}} = \underbrace{\left(1 - \frac{4c_3 + 56}{c_1 + c_2 + 6c_3 + 150}\right)}_{<1} N_{\mathrm{MPF}}. \tag{13}$$

From (13) it is clear that for a given computational complexity more particles can be used in the MPF than in the standard PF. Expression (13) is a specific instance of what has been plotted in Figure 1, where the curve corresponds to $m = 6$, $k = 4$. In order to quantify this statement numerical values for the three constants c_1, c_2 and c_3 are needed. They are estimated by analyzing the actual computational time consumed by various parts of the MPF algorithm. It was fairly easy to measure the time used for likelihood calculation, resampling and random number generation as a function of the number of particles. The problem is to relate them to the

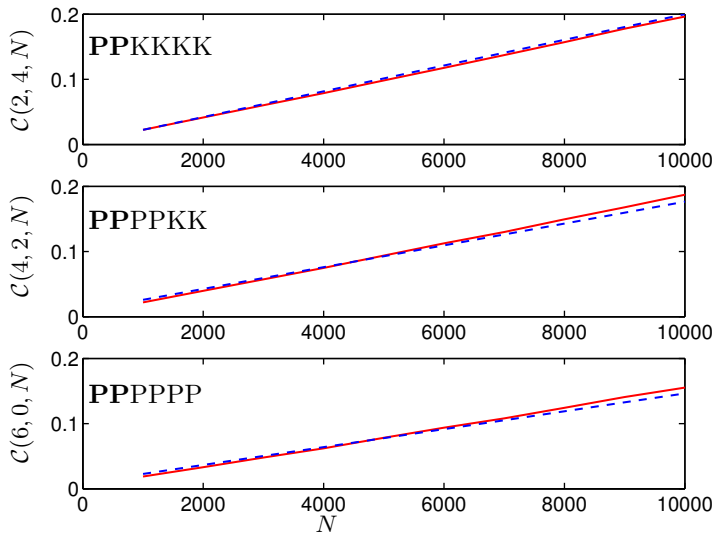


Figure 3: *Using a constant number of particles the times predicted from the theoretical results are shown by the dashed line. The solid line corresponds to the actual time measured using* MATLAB. *If a certain state variable is estimated using the PF this is indicated with a P, and if the KF is used this is indicated using a K.*

time consumed for a single flop. For simpler hardware implementations one flop would have a constant execution time. However, in order to do this on a normal desktop computer running MATLAB, the EF estimation has to be considered, since flop count does not entirely reflect the actual computational time. This is due to memory caching, pipelining, efficient computational routines which are problem size dependent and memory swapping. For the tracking example from (12) the estimated coefficients are $c_1 = 445, c_2 = 487$ and $c_3 = 125$ (on a Sun Blade 100 with 640 MB memory).

By comparing the EF complexity given by (10) to the actual computational time measured in MATLAB it is clear that the predictions of the computational complexity based on theoretical considerations are quite good indeed. The result is given in Figure 3. The small error is mainly due to the fact that it is quite hard to predict the time used for matrix operations, as previously discussed.

4.2 Simulation - Constant Time

Using a constant time the number of particles that can be used is computed. The study is performed by first running the full PF and measure the time consumed by

the algorithm. A MC simulation, using $N = 2000$ particles, is performed in order to obtain a stable estimate of the time consumed by the algorithm. To avoid intervention from the operating system the minimum value is chosen. The time is then used as the target function for the different partitions in the MPF. To find the number of particles needed a search method is implemented and MC simulations are used to get a stable estimate. In Table 2 the number of particles (N), the *root mean square error* (RMSE) and simulation times are shown for the different marginalization cases. RMSE is defined as $\left(\frac{1}{T_f}\sum_{i=1}^{T_f}\frac{1}{N_{MC}}\sum_{j=1}^{N_{MC}}\|x_i^{\text{TRUE}}-\hat{x}_i^{(j)}\|_2^2\right)^{1/2}$, where T_f is the number of time samples and $N_{MC} = 100$ is the number of MC simulations used. From Table 2 it is clear that the different MPFs can use more particles

Table 2: *Results from the constant time simulation.*

	PPPPPP	**PP**KKPP	**PP**PPKK	**PP**KKKK
N	2000	2029	1974	2574
RMSE pos	7.10	5.81	5.76	5.60
RMSE vel	3.62	3.27	3.28	3.21
RMSE acc	0.52	0.47	0.45	0.44
Time	0.59	0.58	0.57	0.60

for a given time, which is in perfect correspondence with the theoretical result given in (13). From the study it is also concluded that the RMSE is decreasing when marginalization is used. This is also in accordance with theory, which states that the variance should decrease or remain unchanged when marginalization is used [8]. Furthermore, Table 2 verifies the theoretical results presented in Figure 1. From this figure it is also clear that the complete marginalization ($m = 6, k = 4$) gives $N(k)/N_0 = 1.44$. Hence, the theoretically predicted number of particles is $2000 \times 1.44 = 2880$. This is in quite good agreement with the result reported in Table 2, 2574.

4.3 Simulation - Constant Velocity RMSE

In this section it is studied what happens if a constant velocity RMSE is used. First the velocity RMSE for the full PF is found using a MC simulation. This value is then used as a target function in the search for the number of particles needed by the different MPFs. Table 3 clearly indicates that the MPF can obtain the same RMSE using fewer particles. The result is that using full marginalization only requires 14% of the computational resources as compared to the standard PF in this example.

Table 3: *Results using a constant velocity RMSE.*

	PPPPPP	**PP**KKPP	**PP**PPKK	**PP**KKKK
N	2393	864	943	264
RMSE pos	7.07	6.98	7.12	7.27
RMSE vel	3.58	3.60	3.65	3.61
RMSE acc	0.50	0.51	0.49	0.48
Time	0.73	0.26	0.28	0.10

5 Conclusion

The contribution in this paper is a systematic approach to analyze and partition the marginalized particle filter from a computational complexity point of view. The method is general and can be applied to a large class of problems. To illustrate the idea, a common target tracking problem is analyzed in detail. The complexity analysis is performed theoretically by counting the number of flops and using the equivalent flop measure to account for complex algorithmic parts such as random number generation and resampling. In an extensive Monte Carlo simulation different performance aspects are shown, and the theoretical results are illustrated and validated.

Acknowledgment

This work was supported by VINNOVA's Center of Excellence ISIS (Information Systems for Industrial Control and Supervision), in particular the partner NIRA Dynamics, and by the Swedish Research Council (VR). The authors would also like to thank the reviewers for their constructive comments.

References

[1] C. Andrieu and A. Doucet. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society*, 64(4):827–836, 2002.

[2] C. Andrieu and S. J. Godsill. A particle filter for model based audio source separation. In *ICA 2000*, Helsinki, Finland, June 2000.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[4] G. Casella and C. P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

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

[6] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

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

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

[9] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, 3 edition, 1996.

[10] N. J. Gordon, D. J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[11] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Information and System Sciences. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[12] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the AMSE–Journal of Basic Engineering*, 82:35–45, 1960.

[13] R. Karlsson, T. Schön, and F. Gustafsson. Complexity analysis of the marginalized particle filter. Technical Report LiTH-ISY-R-2611, Department of Electrical Engineering, June 2004.

[14] P-J. Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation*. Linköping Studies in Science and Technology. Licentiate Thesis No. 945, Linköping University, Linköping, Sweden, May 2002.

[15] T. Schön, F. Gustafsson, and P-J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *To appear in IEEE Transactions on Signal Processing*.

Algorithm 1 Marginalized Particle Filter (MPF), $C_t = 0$

1: Initialization: For $i = 1, \ldots, N$, initialize the particles, $x_{0|-1}^{p,(i)} \sim p_{x_0^p}(x_0^p)$ and set $\{x_{0|-1}^{k,(i)}, P_{0|-1}^{(i)}\} = \{\bar{x}_0^k, \bar{P}_0\}$. Set $t = 0$.

2: For $i = 1, \ldots, N$, evaluate the importance weights $\gamma_t^{(i)} = p(y_t|\mathbb{X}_t^{p,(i)}, \mathbb{Y}_{t-1}) = \mathcal{N}(h_t(x_t^{p,(i)}), R_t)$ and normalize $\tilde{\gamma}_t^{(i)} = \frac{\gamma_t^{(i)}}{\sum_{j=1}^N \gamma_t^{(j)}}$.

3: PF measurement update (resampling): Resample N particles with replacement according to,

$$\text{Prob}(x_{t|t}^{p,(i)} = x_{t|t-1}^{p,(j)}) = \tilde{\gamma}_t^{(j)}. \tag{6}$$

4: PF time update and KF update

(a) KF measurement update,

$$\hat{x}_{t|t}^{k,(i)} = \hat{x}_{t|t-1}^{k,(i)},\ P_{t|t} = P_{t|t-1}. \tag{7}$$

(b) PF time update (prediction): For $i = 1, \ldots, N$,

$$x_{t+1|t}^{p,(i)} \sim p(x_{t+1|t}^p|\mathbb{X}_t^{p,(i)}, \mathbb{Y}_t), \tag{8}$$

where

$$p(x_{t+1}^{p,(i)}|\mathbb{X}_t^{p,(i)}, \mathbb{Y}_t) = \mathcal{N}(A_t x_t^{p,(i)} + A_t^k \hat{x}_{t|t}^{k,(i)},\ A_t^k P_{t|t}(A_t^k)^T + Q_t^p).$$

(c) KF time update,

$$\begin{aligned}
\hat{x}_{t+1|t}^{k,(i)} &= F_t^k \hat{x}_{t|t}^{k,(i)} + F_t^p x_t^{p,(i)} + L_t(x_{t+1|t}^{p,(i)} - A_t^p x_t^{p,(i)} - A_t^k \hat{x}_{t|t}^{k,(i)}),\\
P_{t+1|t} &= F_t^k P_{t|t}(F_t^k)^T + Q_t^k - L_t M_t L_t^T,\\
M_t &= A_t^k P_{t|t}(A_t^k)^T + Q_t^p,\\
L_t &= F_t^k P_{t|t}(A_t^k)^T M_t^{-1},
\end{aligned}$$

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

Paper C

Bayesian Surface and Underwater Navigation

Edited version of the paper:

R. Karlsson and F. Gustafsson. Surface and underwater navigation using particle filters. *Submitted to IEEE Transactions on Signal Processing.*

Parts of the paper in:

R. Karlsson, F. Gustafsson, and T. Karlsson. Particle filtering and Cramér-Rao lower bound for underwater navigation. In *Proceedings IEEE Conference on Acoustics, Speech and Signal Processing*, Hong Kong, April 2003.

R. Karlsson and F. Gustafsson. Particle filter for underwater terrain navigation. In *IEEE Statistical Signal Processing Workshop*, St. Louis, USA, October 2003. Invited paper.

R. Karlsson and F. Gustafsson. Using the particle filter as mitigation to GPS vulnerability for navigation at sea. In *Proceedings of IEEE Statistical Signal Processing Workshop*, Bordeaux, France, July 2005. Submitted as invited paper.

R. Karlsson and F. Gustafsson. A system and method for surface navigation using radar and sea map. Swedish patent application. Patent pending SE-0400264-8, February 2004.

Preliminary version published as Technical Report LiTH-ISY-R-2649, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Bayesian Surface and Underwater Navigation

Rickard Karlsson and Fredrik Gustafsson,

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

Abstract

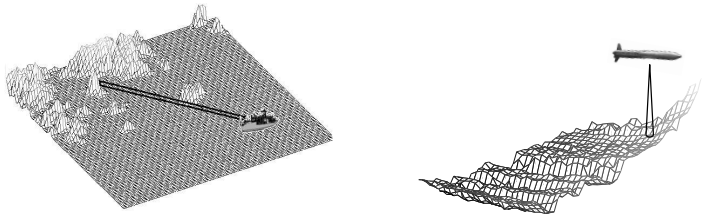
A common framework for surface and underwater (UW) map-aided navigation at sea is proposed, as a supplement to satellite navigation based on the global positioning system (GPS). The proposed Bayesian navigation method is based on information from a distance measuring equipment (DME) and information from databases. For the described system, the fundamental navigation performance expressed as the Cramér-Rao lower bound (CRLB) is analyzed and an analytic solution as a function of the position is derived. As a solution to the recursive Bayesian navigation problem, the particle filter is proposed. Two detailed examples of different navigation applications are discussed: surface navigation using a radar sensor and a sea chart and UW navigation using a sonar sensor and a depth database. In extensive Monte Carlo simulations the performance is shown to be close to the CRLB. The estimation performance for the surface navigation is in comparison with GPS performance. Experimental data is also successfully applied to the UW application.

Keywords: Sea navigation, Recursive Bayesian estimation, Particle filter, Cramér-Rao lower bound.

1 Introduction

Modern sea navigation systems are often based on satellite information from the *global positioning system* (GPS). For critical navigation applications, this sensor cannot be the only positioning sensor. In military applications, an independent backup sensor insensitive to GPS jamming is preferable. Even for civil applications the robustness against jamming may constitute one of the main design issues in the future. In [26, 1] the problem of intentional or unintentional GPS jamming is discussed and alternative backup systems are strongly recommended. This is discussed further in [20] where both bathymetric and celestial methods are described

as alternatives to GPS navigation. At sea, the satellite signal is often received without problem, since there is a free line-of-sight to several satellites. However, under severe weather conditions, such as ice-building on the antenna, or due to the landscape geometry, or a system failure, GPS signals may not be available. There are also applications where the GPS signal is not available at all, for instance in *underwater* (UW) navigation. Hence, there are many reasons to use an independent navigation system.



(a) Surface navigation with radar and a sea chart.

(b) UW navigation using a sonar and a depth map.

Figure 1: *Two DME navigation systems: (a) The surface navigation with a radar sensor and a sea chart database and (b) the UW navigation with a sonar sensor and a depth database.*

Many navigation systems rely on linear models or models that are nearly linear. If the system is approximately described by a Gaussian *probability density function* (pdf) or at least a unimodal density, then the *extended Kalman filter* (EKF), [3, 11], can often be applied successfully. For problems with highly nonlinear or non-Gaussian distributions other methods must be used. One method is to use a bank of several Kalman filters. Several methods are available, for instance IMM, [17]. Also the Gaussian-sum filter, [3], which approximates the underlying pdf by several filters can be used. However, the most general approach is to tackle the nonlinear and non-Gaussian problem directly using the *particle filter* (PF), [8, 7].

As an alternative to satellite navigation it is possible to use terrain information together with sensor information from a *distance measuring equipment* (DME). By comparing the terrain information from a database with the received DME measurements it is in many cases possible to get an accurate position estimate. This map-aided navigation technique is not a new topic, for instance in airborne navigation it has been applied frequently. In [9] a terrain-aided navigation is discussed using EKF; different linearization techniques and multiple models are discussed. In [21] six different stochastic linearization techniques are proposed. In [22, 23] a 3D matching technique using terrain maps is presented. In [27] the positioning is achieved using parallel filters. In [4] a Bayesian terrain navigation problem is solved using a numerical integration directly for the Bayesian recursions. In [5, 7] also a particle filter based method was proposed for aircraft terrain navigation. This was further investigated in [2]. For UW navigation mainly map-matching techniques

or Kalman filter based techniques are previously used. See for instance [19], where a map-matching technique is described for positioning, and [6], where a Kalman filter and data association method is applied to sea floor map-aided navigation.

In this paper, the main focus is on the particle filter as the proposed solution to the recursive Bayesian navigation problem and on analysis of navigation performance, mainly using the *Cramér-Rao lower bound* (CRLB). A common framework for both surface and UW navigation using map-aided navigation together with DME information is formulated. A novel map-aided DME navigation system that does not require any external infrastructure and which is insensitive to jamming is proposed for surface navigation at sea. Using a radar sensor the range to shore in several directions can be compared to sea chart information, in order to calculate the probability for different locations. To the best of the authors knowledge, this is a completely new technique. Also a UW terrain-aided navigation using a sonar sensor is described, based on a similar technique as the airborne navigation system from [5] and using the preliminary results from [14, 13]. For the UW navigation a detailed depth map is compared to depth information retrieved from a sonar sensor in a similar way. For both applications range information from the DME and data from the movements, for instance speed and heading, are used to calculate the position. In a statistical framework the pdf for the ship's position is calculated. This can be achieved by considering several possible positions and for each one calculate the probability, using the particle filter. In Figure 1 two navigation applications are illustrated.

The paper is organized as follows: In Section 2 the navigation system is described. In Section 3 its fundamental performance is discussed using the CRLB. In Section 4 the Bayesian estimation problem is formulated and the approximate solution using the particle filter is described. In Section 5 the novel surface navigation method using a sea chart and a radar sensor is presented. Also the UW navigation based on sonar measurements and a depth database, [14, 13] is presented. Both extensive Monte Carlo simulations and experimental data are used. Finally, in Section 6 conclusive remarks are given.

2 Navigation Models

In this section the common model used in the surface navigation and UW navigation system is presented. Both the system dynamics and the measurement relation are discussed.

2.1 Motion Model

Depending on the configuration, different sensors can be used, such as speedometers and accelerometers. Hence, the motion can be modeled using as many position derivatives as desired. Here, only longitudinal and lateral motion is considered, where the speed is measured. Consider the following state variables: Cartesian position (X, Y), and crab angle, δ, that is the angle between the velocity vector

and the stem of the ship,

$$x_t = \begin{pmatrix} X_t & Y_t & \delta_t \end{pmatrix}^T, \tag{1}$$

as depicted in Figure 2. The following discrete time model with sample time T is used for the navigation system

$$x_{t+1} = f(x_t, u_t, w_t) = \begin{pmatrix} X_t + v_t T \sin(\varphi_t - \delta_t) \\ Y_t + v_t T \cos(\varphi_t - \delta_t) \\ \delta_t \end{pmatrix} + w_t, \tag{2}$$

with input signal $u_t = \begin{pmatrix} v_t & \varphi_t & \theta_t & \phi_t \end{pmatrix}^T$, consisting of speed, v_t, compass, φ_t, elevation angle, θ_t and azimuth angle, ϕ_t relative to the ship's stem. The sensor azimuth angle, ϕ_t, and elevation angle, θ_t, are not present in the dynamic model, but will be used in the measurement relation described in Section 2.2. The process noise, w_t, is considered independent and describes the model uncertainty. Note that here, the known input signal is in practice values obtained from sensors. Hence, if they are considered as noisy measurements, the process noise will also describe the uncertainty in the input. The symbols used in the navigation model is listed in Table 1. If δ is negligible or known the model simplifies even further to only position states. More advanced dynamic models can also be used, for instance a coordinated turn model, which was the case in [13]. For an overview of possible motion models, see the survey in [18].

Table 1: *Description of symbols for the navigation system.*

Signal	**Symbol**	**Description**
States	X_t	Position (in east direction)
	Y_t	Position (in north direction)
	δ_t	Crab angle
Inputs	φ_t	Compass angle
	v_t	Speed
	ϕ_t	Sensor azimuth angle relative to vessel
	θ_t	Sensor elevation relative to vessel
Measurements	r_t	Radar range or sonar depth
Noise	w_t	Process noise
	e_t	Measurement noise

2.2 Measurement Model

For the surface navigation the range to land objects is measured by a radar sensor, and for the UW navigation the range to the sea floor is measured by a sonar sensor. Both sensors measure distance in the direction of the sensor. Hence, the measurement relation is given by

$$y_t = h(x_t, u_t) + e_t = r(x_t, \phi_t^N, \theta_t) + e_t, \tag{3}$$

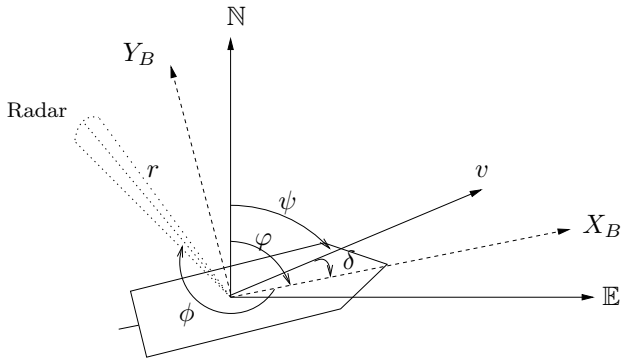


Figure 2: *The ship with compass angle φ defined from the north direction ($\mathbb{N}$), together with the ship's body coordinate frame (X_B, Y_B), crab angle δ, velocity v and heading ψ. The radar lobe is given with azimuth angle ϕ and the range $r = r(X, Y, \phi^N)$, where $\phi^N = \varphi + \phi$.*

where $r(x_t, \phi_t^N, \theta_t)$ is the measured range from position x_t and with the sensor azimuth angle, $\phi_t^N = \varphi_t + \phi_t$, relative to north and with elevation angle θ_t. For the surface navigation the radar angle is defined in Figure 2.

The DME cases for sea navigation is presented in Table 2. Four different sensor types are discussed. The radar uses only azimuth information, hence $\theta_t = 0$. The sonar system uses a fixed elevation. Also the GPS sensor can be viewed as a DME sensor, measuring range for a given position. Finally, the celestial sensor return a binary value. For all the cases different databases are used, for instance a sea chart database, a depth database, a satellite motion database and a star position database.

Table 2: *Special cases of Distance Measuring Equipment (DME).*

Angles	**Distance**	**Sensor**	**Database**
$\theta_t = 0$	$r(x_t, \phi_t^N)$	Radar	Sea chart
$\theta_t < 0$	$r(x_t, \phi_t^N, \theta_t)$	Sonar	Depth map
$\theta_t > 0$	$r(x_t)$	GPS	Satellite information
$\theta_t > 0$	$r(x_t)$	Celestial	Star information

2.3 Navigation System

In Figure 3 the complete surface or underwater navigation system, together with way-point calculation and auto pilot, is depicted. Depending on the application, one or more of the presented sensors are available. For surface navigation, the GPS signal can in many cases be the prime navigation sensor, where the estimate from

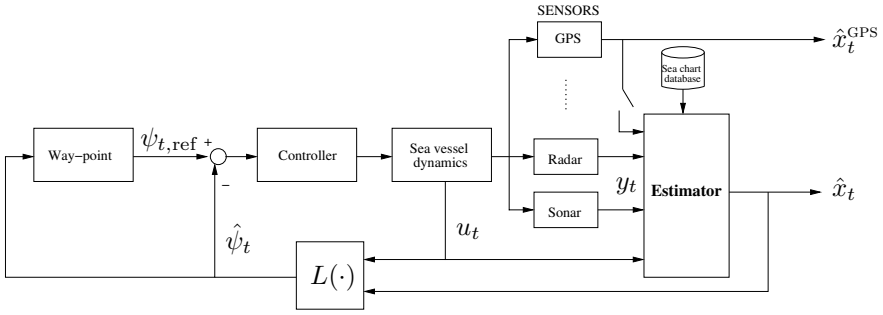


Figure 3: *The complete navigation system, used in the applications in Section 5, consisting of a way-point unit, a controller, the sea chart database, the estimator and various sensors. The reference is given by the desired heading, $\psi_{t,\text{ref}}$ and the estimated heading is given as $\hat{\psi}_t = L(\hat{x}_t, u_t)$.*

the map-aided navigation method can be used to monitor performance. The main objective in this paper is to analyze the map-aided navigation method, hence the GPS part is not further discussed.

3 The Cramér-Rao Lower Bound

The main objective in this section is to find fundamental limits for the estimation performance expressed in terms of the system properties, for instance as a function of the measurement noise or the information available in the sea chart. The *Cramér-Rao lower bound* (CRLB) is such a characteristic for the second order moment [16]. This is done considering a static and dynamic case respectively. In the first case, only the measurement model is considered. In the second, the complete system dynamics is analyzed. The idea in this section is to perform a local analysis in each point in the sea chart and use that to analyze the global behavior.

In the sequel, the CRLB analysis is heavily based on expressions involving gradients of scalar functions or vector valued functions:

$$\nabla_x g(x) = \begin{pmatrix} \frac{\partial g}{\partial x_1} \\ \vdots \\ \frac{\partial g}{\partial x_n} \end{pmatrix}, \; g : \mathbb{R}^n \mapsto \mathbb{R}, \tag{4a}$$

$$\nabla_x g^T(x) = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_m}{\partial x_1} \\ \vdots & & \vdots \\ \frac{\partial g_1}{\partial x_n} & \cdots & \frac{\partial g_m}{\partial x_n} \end{pmatrix}, \; g : \mathbb{R}^n \mapsto \mathbb{R}^m. \tag{4b}$$

Also, the Laplacian for the scalar function $g(x, y)$ with $x \in \mathbb{R}^n, y \in \mathbb{R}^m$ is defined

as

$$\Delta_y^x g(x,y) = \nabla_y (\nabla_x g(x,y))^T,\ g : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}. \tag{5}$$

3.1 Static CRLB

For an unbiased estimator, $\mathbb{E}(\hat{x}) = x$, the CRLB, [16], is given by

$$\text{Cov}(x - \hat{x}) = \mathbb{E}\left((x-\hat{x})(x-\hat{x})^T\right) \succeq J^{-1}(x), \tag{6a}$$
$$J(x) = \mathbb{E}\left(-\Delta_x^x \log p(y|x)\right), \tag{6b}$$

where $J(x)$ is the *Fisher information* matrix and y the measurement. Also note that an equivalent representation of the information, [16], is

$$J(x) = \mathbb{E}\left(\nabla_x \log p(y|x)(\nabla_x \log p(y|x))^T\right). \tag{7}$$

Particularly, for $y = h(x) + e,\ e \in \mathcal{N}(0, R)$ the information is

$$J(x) = H^T(x) R^{-1} H(x), \tag{8}$$

where

$$H^T(x) = \nabla_x h^T(x). \tag{9}$$

Above it is assumed that $J(x)$ is invertible. If not, a local analysis can be done by averaging information around x, in order to get a full rank information matrix. For the case with several independent measurements $y^{(i)}, i = 1, \ldots, M$, this is usually not a problem. The information is then given by

$$J(x) = \sum_{i=1}^{M} J^{(i)}(x), \tag{10}$$

due to the additivity of information, assuming that $J^{(i)}$ is the information in measurement i.

3.2 Dynamic CRLB

The theoretical posterior CRLB for a general dynamic system was derived in [25, 24, 5, 7]. The general state space model can be used to derive fundamental limits for navigation performance. In many applications, δ_t is small and can be discarded from the analysis, or it is known. Hence, for the CRLB analysis the following simplified model is used

$$x_{t+1} = x_t + u_t + w_t, \tag{11a}$$
$$y_t = h(x_t, u_t) = r(x_t, \phi_t^N) + e_t, \tag{11b}$$

where the horizontal position state vector, $x_t \in \mathbb{R}^2$, and the input signal, u_t, are defined as

$$x_t = \begin{pmatrix} X_t \\ Y_t \end{pmatrix}, \; u_t = \begin{pmatrix} v_t \sin(\varphi_t) \\ v_t \cos(\varphi_t) \end{pmatrix}, \tag{12}$$

assuming independent additive process noise w_t. The observation relation consists of range measurements with measurement noise e_t. Using standard notations, consider independent noise sources, with variances $Q_t = \mathrm{Cov}\,(w_t)$ and $R_t = \mathrm{Cov}\,(e_t)$, the posterior CRLB for filtering can be written as, [5]

$$P_{t+1|t+1}^{-1} = Q_t^{-1} + J_{t+1} - S_t^T \left(P_{t|t}^{-1} + V_t \right)^{-1} S_t, \tag{13}$$

where

$$V_t = \mathbb{E}\left(-\Delta_{x_t}^{x_t} \log p(x_{t+1}|x_t) \right), \tag{14a}$$
$$S_t = \mathbb{E}\left(-\Delta_{x_t}^{x_{t+1}} \log p(x_{t+1}|x_t) \right), \tag{14b}$$
$$Q_t^{-1} = \mathbb{E}\left(-\Delta_{x_{t+1}}^{x_{t+1}} \log p(x_{t+1}|x_t) \right), \tag{14c}$$
$$J_t = \mathbb{E}\left(-\Delta_{x_t}^{x_t} \log p(y_t|x_t) \right), \tag{14d}$$

where the bound is given by

$$\mathrm{Cov}\left(x_t - \hat{x}_{t|t} \right) = \mathbb{E}\left((x_t - \hat{x}_{t|t})(x_t - \hat{x}_{t|t})^T \right) \succeq P_{t|t}. \tag{15}$$

If the model in (11) is considered with Gaussian process noise, it gives $S_t = V_t = Q_t^{-1}$. Hence, given the true trajectory for the state vector the posterior bound can be calculated.

In order to analyze the performance, without performing any simulations, a local analysis of the dynamic system is conducted. The assumption is that the covariance should reach a stationary value, i.e., consider a point x^0 and assume $\bar{P}(x) = P_{t+1|t+1}(x) = P_{t|t}(x)$, for all x such that $|x - x^0| < \epsilon$. Using this assumption in (13), it is possible to obtain an expression for $\bar{P}(x)$. This is done by applying the *matrix inversion lemma*

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1}, \tag{16}$$

using $A = Q, B = D = \mathcal{I}$ and $C = \bar{P}(x)$, where $\mathcal{I}$ is the identity matrix. This gives

$$\bar{P}^{-1}(x) = \left(\bar{P}(x) + Q \right)^{-1} + J(x), \tag{17}$$

Solving for $J(x)$ and applying the matrix inversion lemma again with $A = B = D = \bar{P}(x)$ and $C = Q^{-1}$ yield

$$J(x) = \bar{P}^{-1}(x) - \left(\bar{P}(x) + Q \right)^{-1} = \left(\bar{P}(x) + \bar{P}(x)Q^{-1}\bar{P}(x) \right)^{-1}. \tag{18}$$

Hence, if J is assumed invertible, then

$$\bar{P}(x) + \bar{P}(x)Q^{-1}\bar{P}(x) - J(x)^{-1} = 0. \tag{19}$$

Completing squares gives

$$\left(\bar{P}(x)Q^{-1/2} + \frac{1}{2}Q^{1/2}\right)\left(Q^{-1/2}\bar{P}(x) + \frac{1}{2}Q^{1/2}\right) = J(x)^{-1} + \frac{1}{4}Q. \tag{20}$$

Multiply with $Q^{-1/2}$ from left and right in the expression

$$\left(Q^{-1/2}\bar{P}(x)Q^{-1/2} + \frac{1}{2}\mathcal{I}\right)\left(Q^{-1/2}\bar{P}(x)Q^{-1/2} + \frac{1}{2}\mathcal{I}\right) = Q^{-1/2}J(x)^{-1}Q^{-1/2} + \frac{1}{4}\mathcal{I}. \tag{21}$$

Since all matrices are symmetric, a unique matrix square root can be defined. Hence, solving for the symmetric and positive definite matrix, $\bar{P}(x) > 0$, now yields

$$Q^{-1/2}\bar{P}(x)Q^{-1/2} + \frac{1}{2}\mathcal{I} = \left(Q^{-1/2}J(x)^{-1}Q^{-1/2} + \frac{1}{4}\mathcal{I}\right)^{1/2}. \tag{22}$$

Hence, the covariance $\bar{P}(x)$ is given as

$$\bar{P}(x) = -\frac{1}{2}Q + Q^{1/2}\left(Q^{-1/2}J(x)^{-1}Q^{-1/2} + \frac{1}{4}\mathcal{I}\right)^{1/2}Q^{1/2}. \tag{23}$$

One problem with the CRLB using the information form is that the Fisher information, J, may in general be singular. Often because the measurement subspace dimension is much smaller than the state-space dimension. This can be handled using one of the following methods:

(i) Since a global analysis using local behavior is performed, it makes sense to average the information in a neighborhood i.e., $J(x) = \frac{1}{M}\Sigma_{k=1}^{M} J_k(x^{(i)})$, where $x^{(i)}$ are picked around x, so $|x - x^{(i)}| < \epsilon$.

(ii) Introduce more measurements, so $J(x) = H^T(x)R^{-1}H(x)$ becomes invertible.

(iii) A regularization term can be added, basically meaning that a fictitious measurement with a large variance is introduced.

In order to understand the analytic CRLB expression in (23) assume that J is invertible, and consider the special case $Q = q\mathcal{I}$ and $R = r\mathcal{I}$. The covariance $\bar{P}(x)$ is rewritten in order to perform a Taylor expansion:

$$\bar{P}(x) = \begin{cases} -\frac{q}{2}\mathcal{I} + \frac{q}{2}\left(\mathcal{I} + 4q^{-1}J(x)^{-1}\right)^{1/2}, \\ -\frac{q}{2}\mathcal{I} + q^{1/2}J^{-1/2}(\mathcal{I} + \frac{q}{4}J)^{1/2}. \end{cases} \tag{24}$$

Introduce the *signal to noise ratio* (SNR) as

$$\mathrm{SNR} \triangleq qJ = \frac{q}{r} H^T H. \tag{25}$$

Hence, for large or small SNR, a Taylor expansion yields

$$\bar{P} \approx \begin{cases} J^{-1}\left(\mathcal{I} - \mathrm{SNR}^{-1}\right), \\ q\left(-\frac{1}{2}\mathcal{I} + \mathrm{SNR}^{-1/2} + \frac{1}{8}\,\mathrm{SNR}^{1/2} - \frac{1}{32}\,\mathrm{SNR}^{3/2}\right). \end{cases} \tag{26}$$

4 Recursive Bayesian Estimation

Navigation problems are often treated as Bayesian inference. The two map aided navigation methods described in Section 2 are described by nonlinear problems. Consider the following general state-space model

$$x_{t+1} = f(x_t, u_t, w_t), \tag{27a}$$
$$y_t = h(x_t) + e_t, \tag{27b}$$

where $x_t \in \mathbb{R}^n$ denotes the state of the system, u_t the input signal and y_t the observation at time t. The process noise w_t and measurement noise e_t are assumed independent with densities p_{w_t} and p_{e_t} respectively. Let $\mathbb{Y}_t = \{y_i\}_{i=0}^t$ be the set of observations until present time.

The Bayesian estimation problem is given by, [10],

$$p(x_{t+1}|\mathbb{Y}_t) = \int_{\mathbb{R}^n} p(x_{t+1}|x_t) p(x_t|\mathbb{Y}_t)\, dx_t, \tag{28a}$$
$$p(x_t|\mathbb{Y}_t) = \frac{p(y_t|x_t) p(x_t|\mathbb{Y}_{t-1})}{p(y_t|\mathbb{Y}_{t-1})}, \tag{28b}$$

where $p(x_{t+1}|\mathbb{Y}_t)$ is the prediction density and $p(x_t|\mathbb{Y}_t)$ the filtering density. The problem is in general not analytically solvable. To solve the non-tractable Bayesian estimation problem in an on-line application without using linearization or Gaussian assumptions, sequential Monte Carlo methods, or *particle filters* (PF), can be used.

In this section a brief description of the particle filter theory is given. For more details see [5, 7, 8, 12]. The particle filter method provides an approximative Bayesian solution to (28) by approximating the probability density $p(x_t|\mathbb{Y}_t)$ by a large set of N particles $\{x_t^{(i)}\}_{i=1}^N$, where each particle has an assigned relative weight, $\gamma_t^{(i)}$, such that all weights sum to unity. The location and weight of each particle reflect the value of the density in that region of the state space. The likelihood $p(y_t|x_t)$ is calculated from (27) yielding

$$\gamma_t = p(y_t|x_t) = p_{e_t}(y_t - h(x_t)). \tag{29}$$

By introducing a resampling step, as proposed in [8], problems with divergence can be handled. This is referred to as *sampling importance resampling* (SIR), and is summarized in Algorithm 1.

Algorithm 1 Sampling Importance Resampling (SIR)

1: Let $t = 0$. Generate N samples $\{x_0^{(i)}\}_{i=1}^N$ from $p(x_0)$.
2: Compute $\gamma_t^{(i)} = p_e(y_t - h(x_t^{(i)}))$ and normalize, i.e., $\bar{\gamma}_t^{(i)} = \gamma_t^{(i)} / \sum_{j=1}^N \gamma_t^{(j)}$, $i = 1, \ldots, N$.
3: Generate a new set $\{x_t^{(i\star)}\}_{i=1}^N$ by resampling with replacement N times from $\{x_t^{(i)}\}_{i=1}^N$, with probability $\bar{\gamma}_t^{(i)} = \text{Prob}\left(x_t^{(i\star)} = x_t^{(i)}\right)$.
4: $x_{t+1}^{(i)} = f(x_t^{(i\star)}, u_t, w_t^{(i)})$, $i = 1, \ldots, N$ using different noise realizations, $w_t^{(i)}$.
5: Increase t and iterate to step 2.

5 Applications

In this section, two specific applications of our general framework are presented:

- Surface navigation based on range information from the ship's radar and a sea chart.
- UW navigation based on sonar depth measurements and a detailed depth database.

For both applications Gaussian process noise and measurement noise are assumed. With more detailed motion models or sensor models other assumptions are possible.

5.1 Surface Navigation

In this section the novel navigation algorithm is described. The navigation system is presented in Figure 3, where the ship is equipped with a radar, measuring relative distance to any land object. The sensor is assumed stabilized or that the deviation angle is small relative to the uncertainty in the radar sensor. To simplify the analysis, the speed relative to ground and the compass angle are considered as input signals, i.e., noiseless measurements, as described in Section 2.1. However, this assumption is not necessary, and by introducing them as states-variables in the model, these can be estimated. Crucial for the positioning algorithm is a comparison of relative range measurements from the radar with expected land areas from a sea chart database. This is done in a statistically optimal way, using the particle filter as described in Section 4. The map presented in Figure 4 is computed from a vectorized sea chart.

Since the speed is rather small compared to the radar revolution, a complete radar picture can be processed for each time the filter is updated, Figure 5. In order to reduce the amount of possible range data, only a fixed number, (M), of radar strobes are considered each revolution. The radar produces many range measurements in any given direction. In the algorithm, only the measurement closest to the ship is considered. The database consists of a sea chart, in the sequel a portion of the Baltic Sea is studied. In Figure 4 the scenario is presented, where the ship's true position and some radar measurements are depicted. Also,

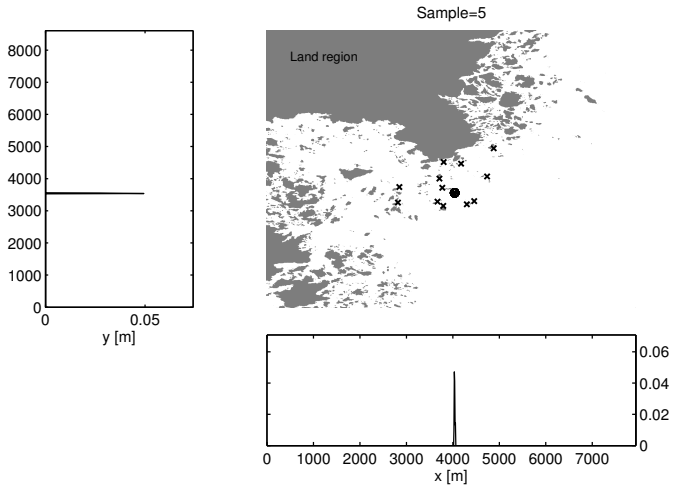


Figure 4: *The true position (o) and radar measurements (x), together with the particle cloud. The marginalized pdf for the X and Y directions are also shown.*

the probability density function for each coordinate is given. As seen, after only 5 revolutions of the radar an accurate position is given. The initial distribution for the example given in Figure 4 was uniformly distributed around the true position, ± 800 m in each direction. The radar measurements from Figure 4 presented in a polar plot with the ship in the origin is presented in Figure 5.

In a Monte Carlo simulation study the scenario given previously is used. A straight trajectory is used in all simulations with different measurement noise realizations. In Figure 6 the RMSE as a function of time is given for 50 Monte Carlo simulations. The parameters used in the Monte Carlo evaluation are presented in Table 3. As seen in Figure 6 the RMSE from the Monte Carlo simulations are close the fundamental limit. The performance using the map-aided navigation method equals that or is slightly better than of an ordinary GPS based navigation system, when there are sufficient returns from land objects. Also note that the peak is due to a region when very few of the radar strobes (from the downsampled radar picture) actually reflect any land area. Now M different strobes are used each radar revolution, so even if land areas are present in the radar picture, sometimes no value is available in the downsampled set. Another possibility is that always M different values are used. The theoretical CRLB calculation from Section 3 can be applied to the surface navigation problem. In Figure 7 each point in the map is allocated the standard deviation $tr\left(\bar{P}(x)\right)$ using (23).

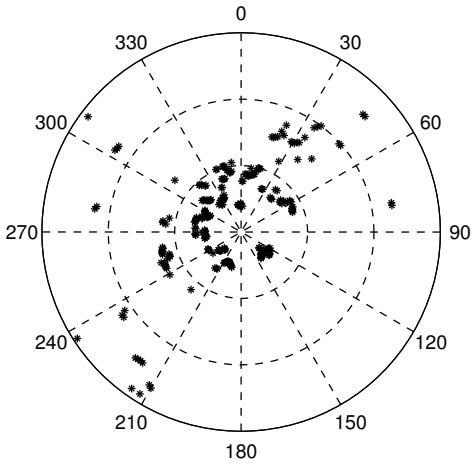


Figure 5: *The radar measurements from one revolution of the radar using a polar representation with the ship's position at the origin. The nearest distance to any object is visualized for all direction with the resolution given by the radar sensor.*

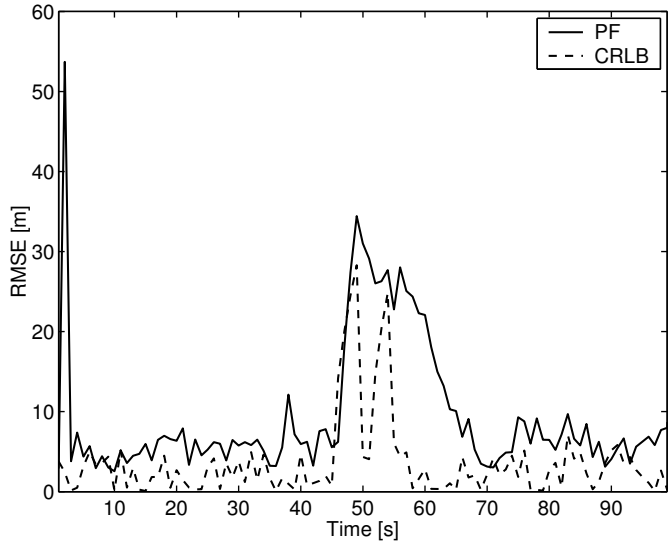


Figure 6: *Surface navigation position RMSE(t) for the particle from* 50 *Monte Carlo simulations together with the CRLB limit as the solution of the EKF around the true trajectory.*

Table 3: *Surface navigation parameters.*

Monte Carlo simulations	50
Process noise covariance	$Q = \text{diag}(10^2,\ 10^2,\ 0^2)$
Measurement noise covariance	$R = 10^2$
Max radar meas./revolution	$M = 16$
Sample time	$T = 1$
Number of particles (initially)	$N = 50000$
Radar interval	$R_{\min} = 300,\ R_{\max} = 6000$

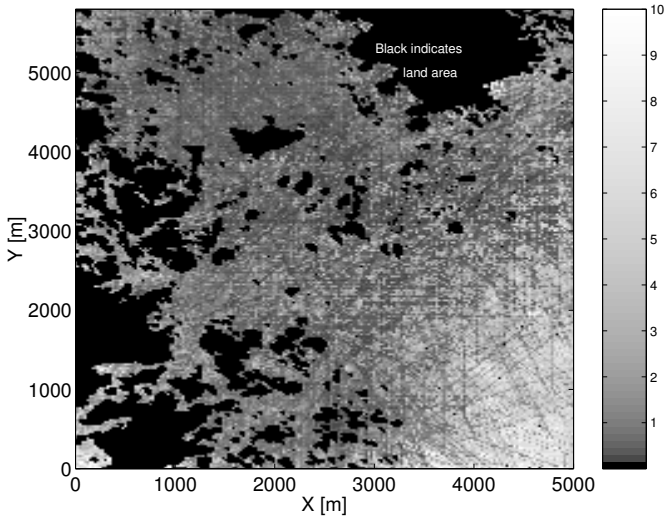


Figure 7: *The analytic CRLB for surface navigation for each position in the sea chart according to the analytic CRLB expression* (23).

5.2 Underwater Navigation

In this section the UW navigation method from [14, 15] is described in detail, using the model from Section 2. In [15] an UW terrain map was collected using sonar depth measurements and differential GPS. In Figure 9 the original data is shown. This data is then interpolated and resampled to get a uniform depth map. This is depicted in Figure 8, together with the platform at depth d_t and with sonar range measurements r_t. In the experiment and in the simulations $d_t = 0$. After the data for map generation was collected, an independent test run in the map region was done, in order to collect measurements to test the map-aided navigation system. The experimental test run is presented in Figure 10 together with the level curves. If the level curves are studied, one can see that the terrain is sufficiently varied for successful positioning even in the somewhat flat regions. In order to test

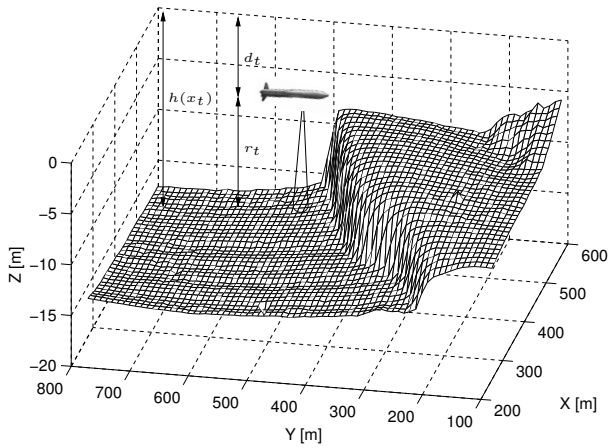


Figure 8: *UW navigation using sonar depth measurements and a true UW terrain database. The depth is d_t, and the sonar indicated the relative range to the sea floor, r_t. The database gives $h(x_t)$.*

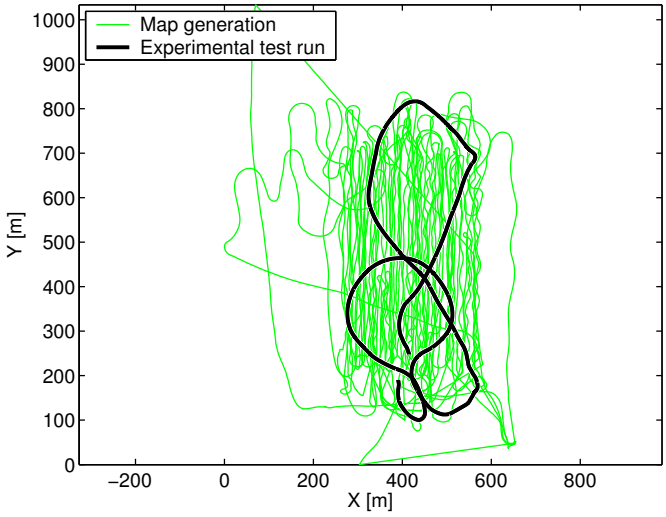


Figure 9: *The trajectory used to record depth data in order to produce the depth map, together with an independent test run on which the particle filter algorithm is tested.*

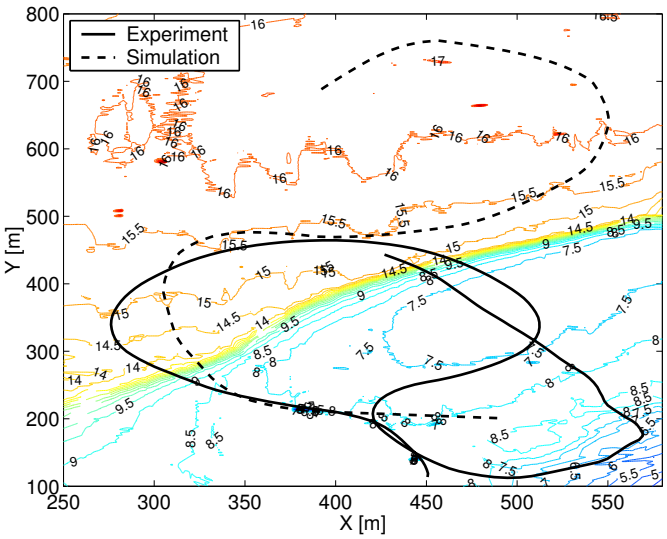


Figure 10: *Part of the experimental test run used for evaluation in the particle filter. The depth map is presented with level curves for each* 0.5 *m level. Also the trajectory used in the Monte Carlo simulation study is visualized.*

the system performance, a Monte Carlo study on simulated data, but using the depth map from the experiment is made. The parameters used in the Monte Carlo evaluation and the experiment are presented in Table 4. The result is presented in

Table 4: *UW navigation parameters.*

Monte Carlo simulations	50
Process noise covariance	$Q = \text{diag}(1^2,\ 1^2,\ 0^2)$
Measurement noise covariance	$R = 0.1^2$
Sample time	$T = 5$
Number of particles (initially)	$N = 50000$

Figure 11. Note that in the upper plot, one experimental test run was used in the RMSE calculation. In the lower plot the RMSE is compared to the CRLB using 50 Monte Carlo simulations. The RMSE is close to the lower bound, and the error is mainly due to deficiencies in the map.

In order to analyze the performance, without doing Monte Carlo simulations, the CRLB technique from Section 3 is used. The problem with the UW navigation is that only one depth measurement is given each time the filter is updated. Hence, the Fisher information, J, is singular, and the theoretical result cannot be applied

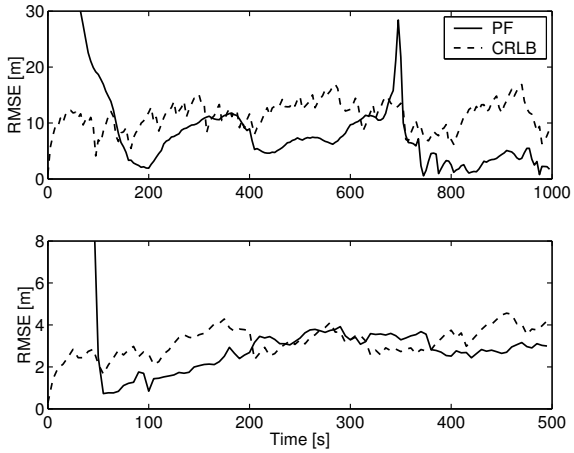


Figure 11: *Estimation error using the trajectories in Figure 10. Upper: The position RMSE for the experimental test run and the CRLB as the EKF solution around the true trajectory. Note that only one test run was used in the RMSE calculation. Lower: The position RMSE from* 50 *Monte Carlo simulations.*

directly. Since the idea behind the analytic CRLB was to use a local analysis of the dynamic system to get a global behavior, it is reasonable to average the information in a neighborhood of each point. Hence, the information matrix becomes invertible. The result from the CRLB calculations using (23) is presented in Figure 12.

6 Conclusions

In this paper a framework for sea navigation using sea chart information and a distance measuring equipment is developed. The navigation performance is analyzed both theoretically and using Monte Carlo simulations. An analytic Cramér-Rao lower bound for the proposed navigation model has been derived. Two specific applications are studied in detail: surface and underwater navigation. The novel surface navigation method uses a radar sensor and a sea chart. In an extensive Monte Carlo simulation the system reaches GPS performance using only a radar sensor and the map-aided navigation method. The underwater navigation uses sonar and a depth map. Both Monte Carlo simulation and experimental data are successfully employed.

Acknowledgment

This work was supported by the VINNOVA's Center of Excellence ISIS (Information Systems for Industrial Control and Supervision) at Linköping University.

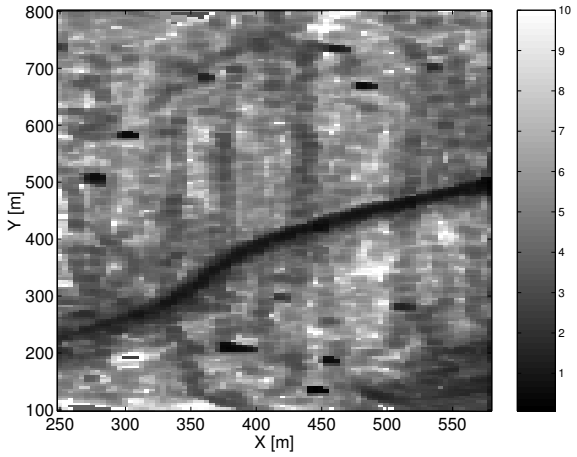


Figure 12: *The analytic CRLB according to* (23)*, for each position in the map in Figure 10.*

The authors would also like to thank: Saab Bofors Underwater Systems, Björn Johansson, Anna Falkenberg and Tobias Karlsson for providing UW topological data, and Björn Gabrielsson for initial discussions and valueful ideas. The Swedish Maritime Administration (Sjöfartsverket) for providing vector based sea chart information. Erik Svensson for assisting with sea chart vector data and map coordinate transformations.

References

[1] GNSS vulnerability & mitigation measures. A study for the European maritime radionavigation forum (rev. 6). Technical report, 2001.

[2] M. Ahlström and M. Calais. Bayesian terrain navigation with Monte Carlo method. Master's Thesis LiTH-ISY-EX-3051, Department of Electrical Engineering. Linköping University, Linköping, Sweden, 2000. In Swedish.

[3] B.D.O. Anderson and J. B. Moore. *Optimal Filtering.* Prentice Hall, Englewood Cliffs, NJ, 1979.

[4] N. Bergman. *Bayesian Inference in Terrain Navigation.* Linköping Studies in Science and Technology. Licentiate Thesis No. 649, Linköping University, Linköping, Sweden, 1997.

[5] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications.* Linköping Studies in Science and Technology. Dissertations No. 579, Linköping University, Linköping, Sweden, 1999.

[6] D. E. di Massa and W. K. Stewart. Terrain-relative navigation for autonomous underwater vehicles. In *Proceedings of the MTS/IEEE Conference OCEANS '97*, volume 1, pages 541 – 546, October 1997.

[7] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[8] N. J. Gordon, D. J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[9] L. Hostetler and R. Andreas. Nonlinear Kalman filtering techniques for terrain-aided navigation. *IEEE Transactions on Automatic Control*, 28(3):315–323, March 1983.

[10] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, 1970.

[11] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Information and System Sciences. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[12] R. Karlsson. *Simulation Based Methods for Target Tracking*. Linköping Studies in Science and Technology. Licentiate Thesis No. 930, Linköping University, Linköping, Sweden, February 2002.

[13] R. Karlsson and F. Gustafsson. Particle filter for underwater terrain navigation. In *IEEE Statistical Signal Processing Workshop*, pages 526–529, St. Louis, MO, USA, October 2003. Invited paper.

[14] R. Karlsson, F. Gustafsson, and T. Karlsson. Particle filtering and Cramér-Rao lower bound for underwater navigation. In *Proceedings IEEE Conference on Acoustics, Speech and Signal Processing*, Hong Kong, April 2003.

[15] T. Karlsson. Terrain aided underwater navigation using Bayesian statistics. Master's Thesis LiTH-ISY-EX-3292, Department of Electrical Engineering. Linköping University, Linköping, Sweden, 2003.

[16] S. Kay. *Fundamentals of Statistical Signal Processing*. Prentice Hall, 1993.

[17] X. R. Li and Y. Bar-Shalom. Design of an interactive multiple model algorithm for air traffic control tracking. *IEEE Transactions on Control Systems Technology*, 1:186–194, 1993.

[18] X. R. Li and V.P. Jilkov. A survey of maneuvering target tracking: Dynamics models. In *Proceedings of SPIE Conference on signal and data processing of small targets*, volume 4048, pages 212–235, April 2000.

[19] L. Lucido, J. Opderbecke, V. Rigaud, R. Deriche, and Z. Zhang. A terrain referenced underwater positioning using sonar bathymetric profiles and multiscale analysis. In *Proceedings of the MTS/IEEE conference OCEANS '96*, volume 1, pages 417–421, September 1996.

[20] F. Pappalardi, S. J. Dunham, M. E. LeBlang, T. E. Jones, J. Bangert, and G. Kaplan. Alternatives to GPS. In *Proceedings of the MTS/IEEE Conference and Exhibition (OCEANS'01)*, volume 3, pages 1452–1459, November 2001.

[21] Y. Pei-Jun, C. Zhe, and J.C. Hung. Performance evaluation of six terrain stochastic linearization techniques for tan. In *Proceedings of the IEEE Aerospace and Electronics Conference*, pages 382 – 388, 1991.

[22] J. Rodriguez and J. K. Aggarwal. Navigation using image sequence analysis and 3-D terrain matching. In *Proceedings on Interpretation of 3D Scenes, Workshop*, pages 200 – 207, 1989.

[23] J. Rodriguez and J. K. Aggarwal. Matching aerial images to 3-D terrain maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1138 – 1149, December 1990.

[24] P. Tichavsky, P. Muravchik, and A. Nehorai. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing*, 46(5):1386–1396, 1998.

[25] H. L. Van Trees. *Detection, Estimation and Modulation Theory*. Wiley, New York, 1968.

[26] J. Volpe. Vulnerability assessment of the transportation infrastructure relying on global positioning system final report. Technical report, U.S. Department of Transportation. National Transportation Systems Center, August 2001.

[27] P. Yubo, Z. Chen, and J. C. Hung. BITAN-II: an improved terrain aided navigation algorithm. In *Proceedings of the 1996 IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, pages 1675–1680, 1996.

Paper D

Bayesian State Estimation of a Flexible Industrial Robot

Edited version of the paper:

> R. Karlsson and M. Norrlöf. Bayesian state estimation of a flexible industrial robot. *Submitted to IEEE Transactions on Control Systems Technology.*

Parts of the paper in:

> R. Karlsson and M. Norrlöf. Bayesian position estimation of an industrial robot using multiple sensors. In *IEEE Conference on Control Applications*, Taipei, Taiwan, September 2004.
>
> R. Karlsson and M. Norrlöf. Position estimation and modeling of a flexible industrial robot. In *16th IFAC World Congress*, Prague, Czech Republic, July 2005. To appear.

Preliminary version published as Technical Report LiTH-ISY-R-2677, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Bayesian State Estimation of a Flexible Industrial Robot

Rickard Karlsson and Mikael Norrlöf

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

Abstract

A sensor fusion technique for state estimation of an industrial robot is presented. By measuring the acceleration at the end-effector, the accuracy of the arm angular position, velocity, and acceleration estimates can be improved. The problem is formulated in a Bayesian estimation framework and two solutions are proposed; one using the extended Kalman filter and one using the particle filter. In an extensive simulation study on a realistic flexible industrial robot, the performance is shown to be close to the fundamental Cramér-Rao lower bound. A significant improvement in position accuracy is achieved using the sensor fusion technique and the method is also proven to be robust to parameter variations in the model.

Keywords: Industrial robot, positioning, estimation, particle filter, extended Kalman filter, Cramér-Rao lower bound.

1 Introduction

Modern industrial robot control is usually based only upon measurements from the motor angles of the manipulator. However, the ultimate goal is to move the tool according to a predefined path. In [9] a method for improving the absolute accuracy of a standard industrial manipulator is described. The improved accuracy is achieved through identification of unknown or uncertain parameters in the robot system, and applying the *iterative learning control* (ILC) method, [2, 24], using additional sensors to measure the actual tool position. The aim of this paper is to evaluate the Bayesian estimation techniques for sensor fusion and to estimate the tool position from indirect measurements such as the acceleration at the end-effector. It is assumed that a high accuracy at the tool position is needed, for instance laser cutting, and that low cost sensors such as accelerometers are used to improve positioning. The methods presented are applied to a realistic flexible

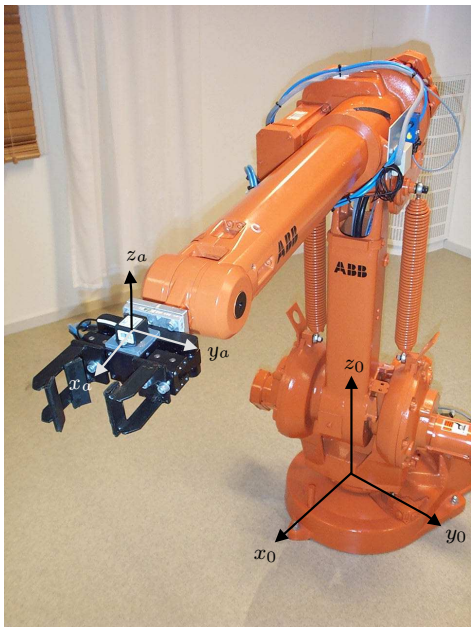


Figure 1: *The ABB IRB1400 robot with an accelerometer mounted at the end-effector. The base coordinate system, (x_0, y_0, z_0), and the coordinate system for the accelerometer, (x_a, y_a, z_a), are also shown.*

robot model and configuration of the system with the accelerometer is depicted in Figure 1.

Traditionally, many nonlinear estimation problems are solved using the *extended Kalman filter* (EKF) [1, 14, 10]. In [11] an EKF is used to improve the trajectory tracking for a rigid 2 *degree-of-freedom* (DOF) robot. Bayesian techniques have been applied in mobile robot applications, see e.g., [21, 13],[6, Ch. 19], but to the best of the authors knowledge, general Bayesian techniques have not yet been applied to position estimation in industrial robotics. The robot dynamics and measurements are highly nonlinear and the measurement noise is not always Gaussian. Hence, linearized models may not always be a good approach. The *particle filter* (PF), [6], provides a general solution to many problems where linearizations and Gaussian approximations are intractable or would yield too low performance. The PF method is also motivated since it provides the possibility to design control laws and perform diagnosis in a much more advanced way. This paper extends the idea introduced in [18]. A performance evaluation in a simulation environment for both the EKF and the PF is presented and it is extensively analyzed using the *Cramér-Rao lower bound* (CRLB) [4, 19]. The sensitivity to model errors is also considered and different levels of model simplification in the measurement equation

are discussed and evaluated.

The paper is organized as follows. In Section 2 the theory of Bayesian estimation is introduced and the equations for the EKF are given. The particle filter algorithm is explained, and the concept of CRLB is presented. The simulation model, the estimation model, and the sensor model, are introduced in Section 3. The results from the simulation experiments are covered in Section 4 and the EKF and PF methods are compared for nominal model parameters but also in a sensitivity analysis with respect to model uncertainty. In Section 5 several motivations for the Bayesian state estimation problem are presented. Finally, Section 6 contains a summary and conclusive remarks.

2 Bayesian Estimation

Consider the discrete state-space model

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

with state variables $x_t \in \mathbb{R}^n$, input signal u_t and measurements $\mathbb{Y}_t = \{y_i\}_{i=1}^t$, with known *probability density functions* (pdfs) for the process noise, $p_w(w)$, and measurement noise $p_e(e)$. The nonlinear prediction density $p(x_{t+1}|\mathbb{Y}_t)$ and filtering density $p(x_t|\mathbb{Y}_t)$ for the Bayesian inference, [12], is given by

$$p(x_{t+1}|\mathbb{Y}_t) = \int_{\mathbb{R}^n} p(x_{t+1}|x_t)p(x_t|\mathbb{Y}_t)dx_t, \tag{2a}$$
$$p(x_t|\mathbb{Y}_t) = \frac{p(y_t|x_t)p(x_t|\mathbb{Y}_{t-1})}{p(y_t|\mathbb{Y}_{t-1})}. \tag{2b}$$

For the important special case of linear-Gaussian dynamics and linear-Gaussian observations the Kalman filter, [15], will give the solution. For nonlinear and non-Gaussian systems, the pdf can not in general be expressed with a finite number of parameters. Instead approximative methods must be used. Usually this is done in two ways; either by approximating the system or by approximating the posterior pdf. See for instance, [30, 3]. Here two different approaches of solving the Bayesian equations are considered; extended Kalman filter, and particle filter. The EKF will solve the problem using a linearization of the system and assuming Gaussian noise. The PF on the other hand will approximately solve the Bayesian equations by stochastic integration. Hence, no linearizations errors occur. The PF can also handle non-Gaussian noise models where the pdfs are known only up to a normalization constant. Also hard constraints on the state variables can be incorporated in the estimation without any problems.

2.1 The Extended Kalman Filter

For the special case of linear dynamics, linear measurements and additive Gaussian noise the Bayesian recursions in Section 2 have an analytical solution, the Kalman

filter. For many nonlinear problems the noise assumptions and the nonlinearity are such that a linearized solution will be a good approximation. This is the idea behind the EKF, [1, 14, 10], where the model is linearized around the previous estimate. Here the time update and measurement update for the EKF is presented,

$$\begin{cases} \hat{x}_{t+1|t} = f(\hat{x}_{t|t}, u_t, 0), \\ P_{t+1|t} = F_t P_{t|t} F_t^T + G_t Q_t G_t^T, \end{cases} \tag{3a}$$

$$\begin{cases} \hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - h(\hat{x}_{t|t-1})), \\ P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}, \\ K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1}, \end{cases} \tag{3b}$$

where the linearized matrices are given as

$$F_t = \nabla_x f(x_t, u_t, 0)|_{x_t = \hat{x}_{t|t}}, \tag{4a}$$

$$G_t = \nabla_w f(x_t, u_t, w_t)|_{x_t = \hat{x}_{t|t}}, \tag{4b}$$

$$H_t = \nabla_x h(x_t)|_{x_t = \hat{x}_{t|t-1}}. \tag{4c}$$

The noise covariances are given as

$$Q_t = \mathrm{Cov}\left(w_t\right), \; R_t = \mathrm{Cov}\left(e_t\right). \tag{5}$$

2.2 The Particle Filter

In this section the presentation of the particle filter theory is according to [4, 6, 7, 16, 28]. The PF provides an approximate solution to the discrete time Bayesian estimation problem formulated in (2) by updating an approximate description of the posterior filtering density. Let x_t denote the state of the observed system and $\mathbb{Y}_t = \{y^{(i)}\}_{i=1}^t$ be the set of observed measurements until present time. The PF approximates the density $p(x_t|\mathbb{Y}_t)$ by a large set of N samples (particles), $\{x_t^{(i)}\}_{i=1}^N$, where each particle has an assigned relative weight, $\gamma_t^{(i)}$, chosen such that all weights sum to unity. The location and weight of each particle reflect the value of the density in the region of the state space, The PF updates the particle location and the corresponding weights recursively with each new observed measurement. For the common special case of additive measurement noise the unnormalized weights are given by

$$\gamma_t^{(i)} = p_e(y_t - h(x_t^{(i)})), \quad i = 1, \ldots, N. \tag{6}$$

Using the samples (particles) and the corresponding weights the Bayesian equations can be approximately solved. To avoid divergence a resampling step is introduced. This is referred to as the *Sampling Importance Resampling* (SIR), [7], and is summarized in Algorithm 1.

Algorithm 1 Sampling Importance Resampling (SIR)

1: Generate N samples $\{x_0^{(i)}\}_{i=1}^N$ from $p(x_0)$.
2: Compute $\gamma_t^{(i)} = p_e(y_t - h(x_t^{(i)}))$ and normalize, i.e., $\bar{\gamma}_t^{(i)} = \gamma_t^{(i)} / \sum_{j=1}^N \gamma_t^{(j)}$, $i = 1, \ldots, N$.
3: Generate a new set $\{x_t^{(i\star)}\}_{i=1}^N$ by resampling with replacement N times from $\{x_t^{(i)}\}_{i=1}^N$, with probability $\bar{\gamma}_t^{(i)} = \text{Prob}(x_t^{(i\star)} = x_t^{(i)})$.
4: $x_{t+1}^{(i)} = f(x_t^{(i\star)}, u_t, w_t^{(i)})$, $i = 1, \ldots, N$ using different noise realizations, $w_t^{(i)}$.
5: Increase t and continue to step 2.

The estimate for each time, t, is often chosen as the minimum mean square estimate, i.e.,

$$\hat{x}_t = \mathbb{E}(x_t) = \int_{\mathbb{R}^n} x_t p(x_t|\mathbb{Y}_t) dx_t \approx \sum_{i=1}^N \gamma_t^{(i)} x_t^{(i)}, \tag{7}$$

but other choices, such as the ML-estimate, might be of interest. The PF approximates the posterior pdf, $p(x_t|\mathbb{Y}_t)$, by a finite number of particles. There exist theoretical limits [6], that the approximated pdf converges to the true as the number of particles tends to infinity.

2.3 Cramér-Rao Lower Bound

When different estimators are used it is fundamental to know the best possible achievable performance. As mentioned previously, the PF will approach the true pdf asymptotically, but for any implementation, due to finite number of particles, only an approximation is given. For other estimators, such as the EKF, it is important to know how much the linearization or model structure used, will affect the performance. The *Cramér-Rao lower bound* (CRLB) is such a characteristic for the second order moment [19]. Here only state-space models with additive Gaussian noise are considered. The theoretical posterior CRLB for a general dynamic system was derived in [32, 31, 4, 6]. Here a continuous-time system is considered. By first linearizing and then discretizing the system, the fundamental limit can in practice be calculated as the stationary solution, $\bar{P} = \bar{P}(x_t^{\text{TRUE}})$, of the Riccati recursions in the EKF, where the linearizations are around the true state trajectory, x_t^{TRUE}. Note that for the industrial robot application a high sample rate and a small process noise make the approximation fairly accurate. The predicted value of the stationary covariance for each time t, i.e., for each point in the state-space, x_t^{TRUE}, is denoted $\bar{P}_p$ and given as the solution to

$$\bar{P}_p = \bar{F}(\bar{P}_p - (\bar{P}_p \bar{H}^T (\bar{H} \bar{P}_p \bar{H}^T + R)^{-1}) \bar{H} \bar{P}_p) \bar{F}^T + \bar{G} Q \bar{G}^T. \tag{8}$$

where the linearized matrices are evaluated around the true trajectory, x_t^{TRUE}. The CRLB limit can now be calculated as

$$\bar{P} = \bar{P}_p - \bar{K} \bar{H} \bar{P}_p, \tag{9}$$

for each point along the state-trajectory.

3 Models

In this section a continuous-time 3 DOF robot model is discussed. The model is simplified and transformed into discrete time where it can be used by the EKF and PF. The measurements are in both cases angle measurements from the motor, with or without acceleration information from the arm.

3.1 Robot Model

A common assumption of the dynamics of the robot is that the transmission can be approximated by two or three masses connected by springs and dampers. The coefficients in the resulting model can be estimated from an identification experiment. See for instance [20]. Here it will be assumed that the transmission can be described by a two mass system and that the manipulator is rigid, schematically shown in Figure 2. The equation describing the torque balance for the motor

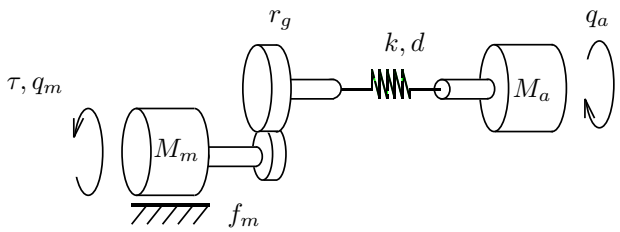


Figure 2: *A two mass model of the dynamics between two joints in the industrial robot; spring (k), damper (d), friction f_m, gear ratio (r_g), moments of inertia (M_m, $M_a(q_a)$), torque (τ) and angles (q_m, q_a).*

becomes

$$\begin{aligned} M_m\ddot{q}_m = &-f_m\dot{q}_m - r_g k(r_g q_m - q_a) \\ &-r_g d(r_g\dot{q}_m - \dot{q}_a) + \tau, \end{aligned} \tag{10}$$

where M_m is the motor inertia matrix, q_m the motor angle, q_a the arm angle, r_g the gear ratio, f_m, k, and d are the motor friction, spring constant and damping respectively. Input to the system is the motor torque, τ. The corresponding relation for the arm becomes a nonlinear equation

$$\begin{aligned} M_a(q_a)\ddot{q}_a + C(q_a, \dot{q}_a)\dot{q}_a + g(q_a) = \\ k(r_g q_m - q_a) + d(r_g\dot{q}_m - \dot{q}_a). \end{aligned} \tag{11}$$

A more detailed model of the robot should include nonlinear friction such as Coulomb friction. An important extension would also be to model the nonlinear spring characteristics in the gear-boxes. In general the gear-box is less stiff for torques close to zero and more stiff when high torques are applied.

An industrial robot has, in general, 6 DOF. However, here only joint 1-3 (not the wrist joints) are used in the simulation study. The continuous-time robot model is implemented in MATLAB Simulink [23], as presented in Figure 3. The manipulator block uses the Simulink block from Robotics Toolbox [5] to simulate the nonlinear dynamics.

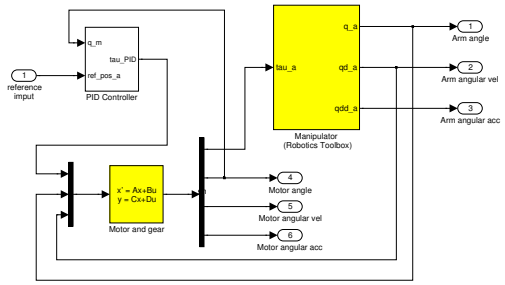


Figure 3: *The* MATLAB *Simulink model of the robot and PID controller.*

The Denavit-Hartenberg (DH) parameters for the robot are given in Table 1. The robot is stabilized using a PID-controller,

$$F_{\text{PID}}(s) = K_{\text{P}} + \frac{K_{\text{I}}}{s} + \frac{K_{\text{D}} s}{\frac{s}{b} + 1}, \tag{12}$$

where s denotes the Laplace operator. The parameters are described in Table 2. For specific values of the dynamics, see [33].

3.2 Estimation Model

The estimation model has to reflect the dynamics in the true system. A straight forward choice of estimation model is the state space equivalent of (11) and (10), this gives a nonlinear dynamic model with 12 states (motor and arm angular positions, velocities). Since the goal is to find an estimate of the arm angles the

Table 1: *DH parameters for the robot used in the simulation.*

Joint/Link i	$\boldsymbol{\alpha}_i$ [rad]	$\mathbf{a}_i$ [m]	$\boldsymbol{\theta}_i$ [rad]	$\mathbf{d}_i$ [m]
1	$-\pi/2$	0.41	0	0.78
2	0	1.075	$-\pi/2$	0
3	$\pi/2$	1.056	$\pi/2$	0

Table 2: *PID parameters used in the simulation.*

Parameter	Joint 1	Joint 2	Joint 3
K_{P}	40	50	40
K_{I}	40	40	40
K_{D}	2	2	2
b	150	150	150

following state variables are used

$$x_t = \begin{pmatrix} q_{a,t} & \dot{q}_{a,t} & \ddot{q}_{a,t} \end{pmatrix}^T, \tag{13}$$

where $q_{a,t} = \begin{pmatrix} q^1_{a,t} & q^2_{a,t} & q^3_{a,t} \end{pmatrix}^T$ contains the arm angles from the first three joints in Figure 1 and $\dot{q}_{a,t}$ is the angular velocity and $\ddot{q}_{a,t}$ is the angular acceleration at time t. This yields the following state space model in discrete time

$$x_{t+1} = F_t x_t + G_{u,t} u_t + G_{w,t} w_t, \tag{14a}$$

$$y_t = h(x_t) + e_t, \tag{14b}$$

where

$$F_t = \begin{pmatrix} \mathcal{I} & T\mathcal{I} & T^2/2\mathcal{I} \\ \mathcal{O} & \mathcal{I} & T\mathcal{I} \\ \mathcal{O} & \mathcal{O} & \mathcal{I} \end{pmatrix}, \tag{15a}$$

$$G_{w,t} = \begin{pmatrix} \frac{T^2}{2}\mathcal{I} \\ T\mathcal{I} \\ \mathcal{I} \end{pmatrix}, \; G_{u,t} = \begin{pmatrix} \frac{T^3}{6}\mathcal{I} \\ \frac{T^2}{2}\mathcal{I} \\ T\mathcal{I} \end{pmatrix}. \tag{15b}$$

This model is linear and the number of states is reduced to 9, compared to the 12 states in the nonlinear state space model. In Section 2.3 a linearized model of the nonlinear dynamics is presented for computing the Cramér-Rao lower bound. The input, u_t, is the arm jerk reference, i.e., the differentiated arm angular acceleration reference. The process noise, w_t and measurement noise e_t are considered Gaussian with zero mean and covariances, Q_t and R_t respectively. The sample time is denoted T and $\mathcal{I}$ and $\mathcal{O}$ are three by three unity and null matrices. The observation relation, (14b), is described in full detail in the next section.

3.3 Sensor Model

The observation relation is given by

$$h(x_t) = \begin{pmatrix} q_{m,t} \\ \ddot{\rho}_t \end{pmatrix}, \tag{16}$$

where $q_{m,t}$ is the measured motor angle and $\ddot{\rho}_t$ is the Cartesian acceleration vector in the accelerometer frame, Figure 1. With the simplified discrete time model

described in Section 3.1, the motor angle $q_{m,t}$ is computed as

$$\begin{aligned} q_m = r_g^{-1}\Big(q_a + k^{-1}(M_a(q_a)\ddot{q}_a + g(q_a) \\ + C(q_a,\dot{q}_a)\dot{q}_a - d(r_g\dot{q}_m - \dot{q}_a))\Big). \end{aligned} \tag{17}$$

The approach is similar to the one suggested in [8], which uses the relation given by (11) in a case when the system is scalar and linear. The results presented here are more general, since a multi-variable nonlinear system is considered.

The kinematics [29] of the robot is described by a nonlinear mapping $\rho_t = \mathcal{T}(q_{a,t})$, and its Jacobian is defined as

$$J(q_a) = \nabla_{q_a}\mathcal{T}(q_a). \tag{18}$$

The following equation relates the Cartesian acceleration with the state variables

$$\ddot{\rho}_t = J(q_{a,t})\ddot{q}_{a,t} + \Big(\sum_{i=1}^{3}\frac{\partial J(q_{a,t})}{\partial q_{a,t}^{(i)}}\dot{q}_{a,t}^{(i)}\Big)\dot{q}_{a,t} + n_g(q_{a,t}), \tag{19}$$

where $q_{a,t}^{(i)}$ is the ith element of $q_{a,t}$ and $n_g(q_{a,t})$ is the gravity vector measured by the accelerometer.

Remark: If the nonlinear dynamics (11) and (10), are used, see Section 3.1, the relation in (17) becomes linear since $q_{m,t}$ is a state variable. However, the relation in (19) becomes more complex since $\ddot{q}_{a,t}$ is no longer a state, but has to be computed using (11).

3.4 CRLB Analysis of the Robot

In Section 2.3 the posterior CRLB was defined for a general nonlinear system with additive Gaussian noise. In this section the focus is on the CRLB expression for the industrial robot presented in Section 3.1. Equation (11) gives

$$\begin{aligned} \kappa(q_a,\dot{q}_a) \triangleq \ddot{q}_a = -M_a(q_a)^{-1}\big(k(r_g q_m - q_a) \\ - d(r_g\dot{q}_m - \dot{q}_a) - g(q_a) - C(q_a,\dot{q}_a)\dot{q}_a\big). \end{aligned} \tag{20}$$

Here, the motor angular velocity, $\dot{q}_m$, is considered as an input signal, hence not part of the state-vector, $x(t) = \begin{pmatrix} q_a & \dot{q}_a & \ddot{q}_a \end{pmatrix}^T$. The system can be written in state space form as

$$\dot{x} = \frac{d}{dt}\begin{pmatrix} q_a \\ \dot{q}_a \\ \ddot{q}_a \end{pmatrix} = f^c(x(t)) = \begin{pmatrix} \dot{q}_a \\ \ddot{q}_a \\ \Lambda(q_a,\dot{q}_a,\ddot{q}_a) \end{pmatrix}, \tag{21a}$$

$$\Lambda(q_a,\dot{q}_a,\ddot{q}_a) = \frac{d}{dt}\kappa(q_a,\dot{q}_q). \tag{21b}$$

The differentiation of κ is performed symbolically, using the MATLAB symbolic toolbox. According to Section 2.3 the CRLB is defined as the stationary Riccati solution of the EKF around the true trajectory, x_t^{TRUE}. This is formulated for a discrete-time system. Hence, the continuous-time robot model from (21) must first be discretized. This can be done by first linearizing the system and then discretize it, [10]. The system is too complicated to apply a symbolic gradient directly in the linearization. However, a numerical differentiation can be done around the true trajectory, yielding

$$A^c = \nabla_x f^c(x)|_{x=x_t^{\text{TRUE}}} = \begin{pmatrix} \mathcal{O} & \mathcal{I} & \mathcal{O} \\ \mathcal{O} & \mathcal{O} & \mathcal{I} \\ \frac{\partial \Lambda(q,\dot{q},\ddot{q})}{\partial q} & \frac{\partial \Lambda(q,\dot{q},\ddot{q})}{\partial \dot{q}} & \frac{\partial \Lambda(q,\dot{q},\ddot{q})}{\partial \ddot{q}} \end{pmatrix}, \tag{22}$$

The desired discrete time system matrix is now given as

$$\bar{F} = e^{A^c \cdot T}, \tag{23}$$

where T is the sample time. In Section 4 the CRLB is compared to the estimation result in Monte Carlo simulations, both with and without accelerations measurements, using the above system and the CRLB expressions presented in Section 2.3.

4 Simulation Results

The model is implemented and simulated using the Robotics Toolbox [5] in MATLAB Simulink as is described in Section 3.1. The simulation study is based mainly around the EKF approach, since it is a fast method well suited for large Monte Carlo simulations. The impact on several different parts in the modeling are studied together with a sensitivity analysis. Also performance simulations around nominal parameters are compared to the CRLB. The PF is much slower, hence a smaller Monte Carlo study is performed. The Monte Carlo simulations use the following covariance matrices for the process and measurement noise

$$Q = 4 \cdot 10^{-6}\mathcal{I}, \quad R = \begin{pmatrix} 10^{-6} \cdot \mathcal{I} & \mathcal{O} \\ \mathcal{O} & 10^{-4} \cdot \mathcal{I} \end{pmatrix}. \tag{24}$$

The system is simulated around the nominal trajectory and produces different independent noise realizations for the measurement noise in each simulation. The same covariances are used in the CRLB evaluation. The continuous-time Simulink model of the robot is sampled in 1 kHz. The data is then decimated to 100 Hz before any estimation method is applied. The arm reference path is presented in Figure 4. The robot illustration is done using Robotics Toolbox, [5], and the path is created using the Path Generation Toolbox (PGT), [27].

4.1 Modeling and Sensitivity Analysis

The model of the angular measurement relation (17) consists of different terms, the angular term, the gravitational term, the inertia term, the Coriolis term and

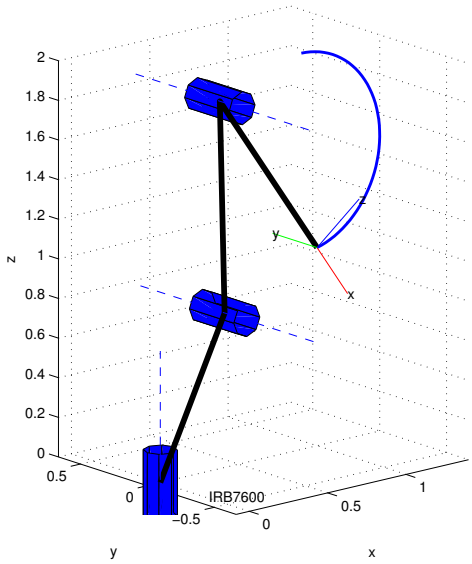


Figure 4: *Illustration of the path in Cartesian space used in the simulation. The configuration of the robot is shown for the initial position of the path.*

the damping term. Since the system does not work properly without any angular measurements this is considered mandatory. The damping part can only be analyzed if $\dot{q}_m$ is available. $\dot{q}_m$ is not part of the state-vector, but could in principle be estimated if included or directly estimated by differentiating the signal q_m. In Figure 5 the first four terms of (17) are shown for the data in the simulation. From Figure 5 the importance of the different terms can be concluded. The term containing q_a is fundamental, the gravitational term is also important since it gives a bias to the estimate. The inertia term also contributes together with the Coriolis term. The damping term has been neglected since it is has a much lower amplitude than the other terms.

The different models are evaluated using the EKF and the error is measured as $\|q_{a,t} - \hat{q}_{a,t}\|_2$. As seen in Figure 6, the incorporation of inertia and Coriolis significantly improve performance. This is also shown in Table 3, where the average error for the noise free case is presented.

To illustrate the estimation idea, the estimate of q_a is presented for the EKF and directly from motor measurements only. The estimate without any model, $\hat{\hat{q}}_a$ is based on

$$\hat{\hat{q}}_a = r_g q_m - k^{-1} g(q_a), \tag{25}$$

where the gravitational part is compensated statically, using the true value. In Figure 7 the estimation and the direct transformation from motor measurements

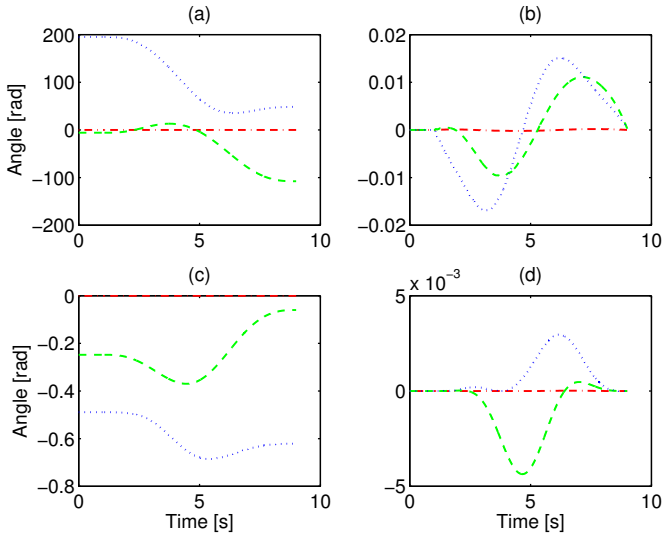


Figure 5: *Terms in the measurement equation,* (17)*, (dash-dotted) represents joint 1, (dashed) joint 2, and (dotted) joint 3 . (a)* $r_g^{-1}q_a$*, (b)* $r_g^{-1}k^{-1}M_a(q_a)\ddot{q}_a$*, (c)* $r_g^{-1}k^{-1}g(q_a)$*, and (d)* $r_g^{-1}k^{-1}C(q_a,\dot{q}_a)\dot{q}_a$*.*

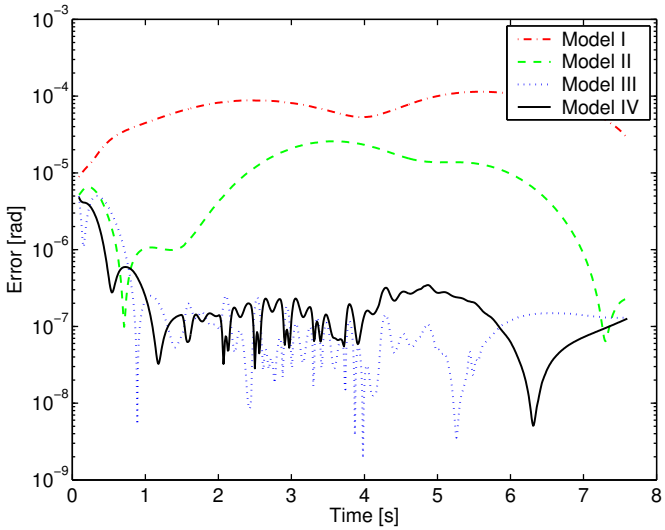


Figure 6: *The angular error,* $\|q_{a,t} - \hat{q}_{a,t}\|_2$*, for different models according to Table 3 using a noiseless realization.*

Table 3: *Average angular estimation error for different models around the nominal trajectory, noise free system.*

Model	Model of measurement eq.	Avg. error [rad]
I.	$r_g^{-1}(q_a + k^{-1}g(q_a))$	$7.66 \cdot 10^{-5}$
II.	$r_g^{-1}(q_a + k^{-1}g(q_a) + k^{-1}M_a(q_a)\ddot{q}_a)$	$1.31 \cdot 10^{-5}$
III.	$r_g^{-1}(q_a + k^{-1}g(q_a) + k^{-1}M_a(q_a)\ddot{q}_a + k^{-1}C(q_a, \dot{q}_a)\dot{q}_a)$	$1.02 \cdot 10^{-6}$
IV.	$r_g^{-1}(q_a + k^{-1}g(q_a) + k^{-1}M_a(q_a)\ddot{q}_a + k^{-1}C(q_a, \dot{q}_a)\dot{q}_a - k^{-1}d(r_g\dot{q}_m - \dot{q}_a))$	$7.53 \cdot 10^{-7}$

only is illustrated based on 500 Monte Carlo simulations. The evaluation uses the *root mean square error* (RMSE) defined as

$$\mathrm{RMSE}(t) = \left(\frac{1}{N_{\mathrm{MC}}} \sum_{j=1}^{N_{\mathrm{MC}}} \| x_t^{\mathrm{TRUE}} - \hat{x}_t^{(j)} \|_2^2 \right)^{1/2}, \tag{26}$$

where N_{MC} is the number of Monte Carlo simulations, x_t^{TRUE} is the true state vector and $\hat{x}_t^{(j)}$ is the estimated state vector in Monte Carlo simulation j. As seen in Figure 7, the EKF gives a significant improvement in arm position even if the accelerometer is not used. This is because the states are available in the EKF and a detailed model can be included. The filter also reduces the effect of measurement disturbances.

It is also interesting to investigate the sensitivity of the estimator for important parameters, such as the damping coefficient d and the spring coefficient k. In Figure 8 the total RMSE for different parameter variations is presented using 10 Monte Carlo simulations for each specific parameter.

4.2 Estimation Performance

EKF. In Figure 9 the RMSE from 500 Monte Carlo simulations are compared with the CRLB limit, both with and without acceleration measurements. The CRLB is computed as the square root of the trace for the covariance matrix part corresponding to the angular states. As seen the RMSE is close the fundamental limit. The discrepancy is due to model errors, i.e., neglected damping term and the fact that the estimator uses a simplified system matrix consisting of integrators only. Also note that the accelerometer measurements reduce the estimation uncertainty. The results in Figure 9 is of course for the chosen trajectory, but the acceleration values are not that large, so greater differences will occur for larger accelerations. The total RMSE, ignoring the initial transient is given in Table 4 for both angular position, velocity and acceleration. The improvement is substantial in angular position, but for control also the improvement in angular velocity and acceleration is

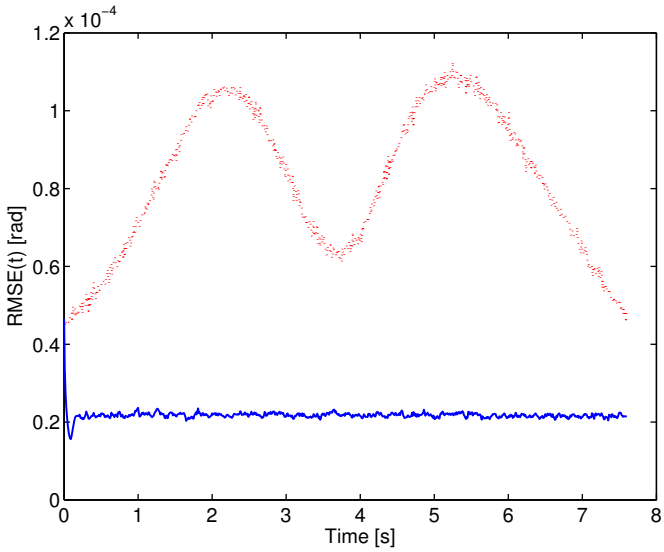


Figure 7: *The arm position estimate, $\hat{q}_a$ from motor-side measurements only (dotted) with static gravity compensation and the estimate $\hat{q}_a$ from EKF without using the accelerometer (solid).*

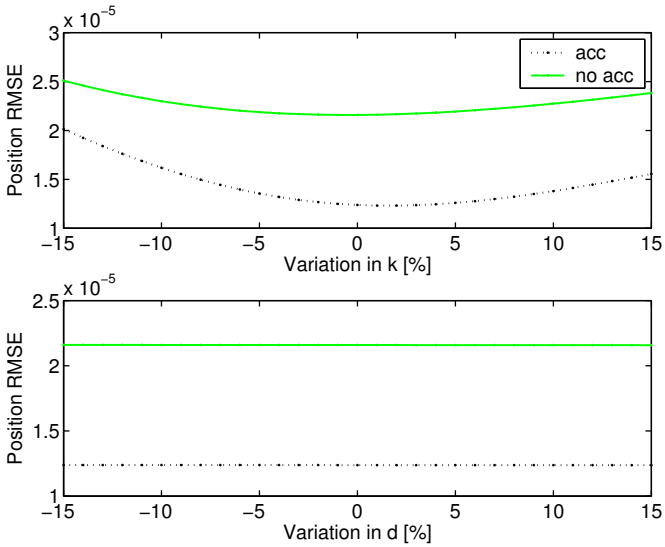


Figure 8: *Total RMSE angular position sensitivity analysis for the spring k variation (upper) and the damper d variation (lower), $N_{MC} = 10$.*

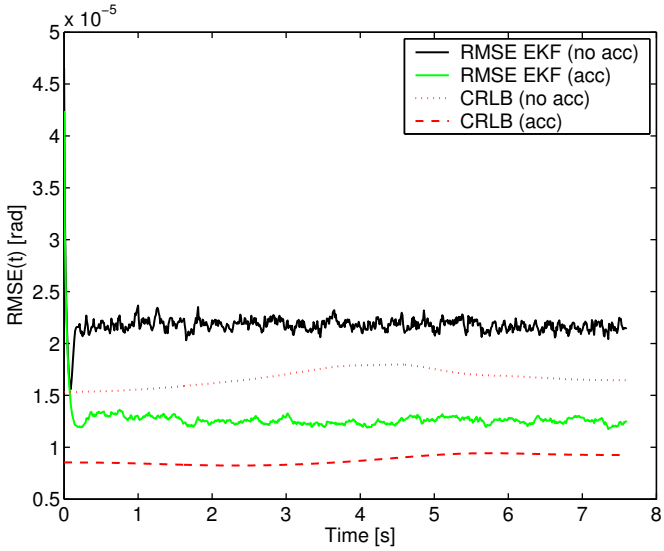


Figure 9: *Angular position RMSE from* 500 *Monte Carlo simulations using the EKF without and with accelerometer sensor are compared to the CRLB limit, i.e., the square root of the trace of the angular position from the CRLB covariance.*

Table 4: *Total RMSE for arm-side angular position (q_a), angular velocity ($\dot{q}_a$) and angular acceleration ($\ddot{q}_a$), with and without the accelerometer, using* 500 *Monte Carlo simulations.*

	Accelerometer	**No accelerometer**
RMSE q_a	$1.25 \cdot 10^{-5}$	$2.18 \cdot 10^{-5}$
RMSE $\dot{q}_a$	$7.57 \cdot 10^{-5}$	$4.08 \cdot 10^{-4}$
RMSE $\ddot{q}_a$	$1.23 \cdot 10^{-3}$	$3.91 \cdot 10^{-3}$

important. On a 1.5 GHz PC running MATLAB the EKF performs in real-time on the 100 Hz data rate.

PF. The particle filter is rather slow compared to the EKF for this model structure. Hence, the given MATLAB implementation of the system is not well suited for large Monte Carlo simulations. Instead a small Monte Carlo study over a much shorter time period than for the EKF case is considered. The PF and the EKF are compared, and a small improvement in performance is noted. The result is given in Figure 10. Even though the PF is slow, it gives more insight in the selection of simulation parameters than the EKF, where the filter performance is more dependent on the ratio between the process and measurement noise. Since

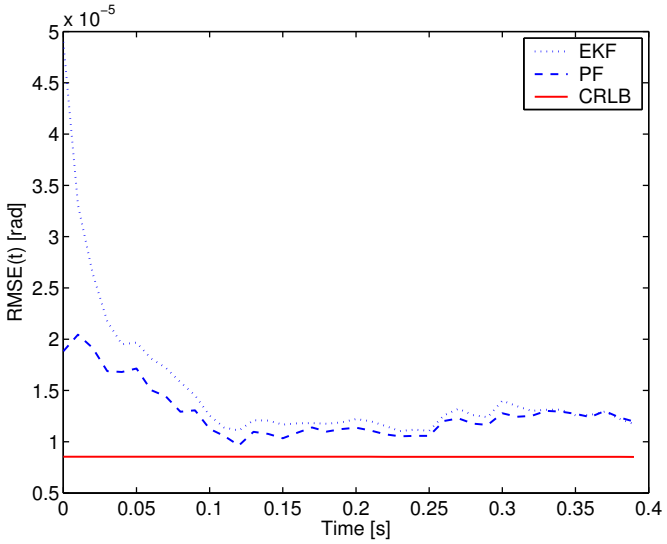


Figure 10: *EKF and PF angular position RMSE with external accelerometer signal from* 20 *Monte Carlo simulations.*

the data rate is rather high the linearization problem is not severe, so the EKF performs sufficiently well. To improve performance a more complicated system dynamics must be implemented in the filter.

5 Motivations

In this section motivations for the Bayesian methods presented are given and future work are indicated. With a highly accurate tool position estimate, the control of the robot can be improved. However, to incorporate the estimates in a closed loop real-time system may not be possible due to the computational complexity in the estimation methods. This is not a problem in many practical applications. Consider for instance ILC, which is an off-line method. ILC has over the years become a standard method for achieving high accuracy in industrial robot control [2, 26, 22]. It utilizes a repetitive system dynamics to compensate for errors. Mathematically an ILC control law can be written as

$$u_{t,k+1} = \mathcal{Q}(u_{t,k} + \mathcal{L}\epsilon_{t,k}), \tag{27}$$

where $u_{t,k}$ is the ILC input in the kth iteration and $\epsilon_{t,k}$ is the error. The error is defined as $\epsilon_{t,k} = r_t - y_{t,k}$ where r is the reference and $y_{t,k}$ the measured output of the system. $\mathcal{Q}$ and $\mathcal{L}$ are design parameters for the control law, often chosen as linear filters [24]. In industrial robot systems the measured output does not correspond

to the actual controlled output. An ILC experiment on the ABB IRB1400 in [25, Chapter 9] using only motor angle measurements, i.e., no accelerometer, shows that although the error on the motor-side is reduced the path on the arm-side does not follow the programmed path. This is illustrated in Figure 11.

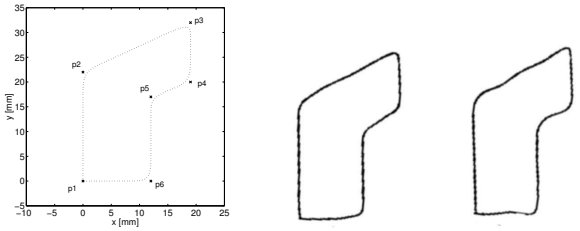


Figure 11: *Results from an ILC experiment on the ABB IRB1400 robot where ILC is applied using only motor angle measurements. Programmed path (left), iteration 0 (middle), and iteration 10 (right). The path on the arm-side does not follow the programmed trajectory.*

The idea in this paper is to use an accelerometer at the end-effector to get measurements that reflects the actual tool motion, see Figure 1. From these measurements and a model of the robot the estimated position error, $\hat{\epsilon}_{t,k}$, is used in the ILC update equation according to

$$u_{t,k+1} = \mathcal{Q}(u_{t,k} + \mathcal{L}\hat{\epsilon}_{t,k}). \tag{28}$$

Using the EKF, $\hat{\epsilon}_{t,k}$ represent the mean error, with an estimate of the covariance from the EKF. Hence, this can be used in the improvement process and an idea in this direction is presented in [26]. The covariance could be used to change the gain of the learning operator $\mathcal{L}$ in order to reduce the effect of random disturbances. In [8] a 1 DOF lab-process is controlled using (28) but the estimation is simplified due to the inherent linearity of the system. Using the PF, the pdf $p(x_t|\mathbb{Y}_t)$ is given by samples, $x_t^{(i)}, i = 1, ..., N$ as described in Section 2.2. Hence, the error consists of samples $\epsilon_t^{(i)}$, so the ILC improvement can be done in a more sophisticated way. The mean estimate or the maximum likelihood (ML) estimate, or a combination thereof are logical choices.

The extra knowledge of the pdf from the PF can also be used in for instance diagnosis. A general model-based statistical decision rule based on a criterion, $g(x_t)$, can be formulated. The probability can be calculated, for instance $\text{Prob}(g(x_t) > 0) > 1 - \beta$, where β determines the confidence level. Here, the samples $x_t^{(i)}$, $i = 1, 2, \ldots, N$ from the distribution $p(x_t|\mathbb{Y}_t)$ can be used to calculate the probability as

$$\text{Prob}(g(x_t) > 0) \approx \frac{\#g(x_t^{(i)}) > 0}{N}, \tag{29}$$

by simply counting the number of samples fulfilling the criteria. Since these samples are directly available in the PF method it is well suited for hypothesis testing. Similar for the EKF, an off-line Monte Carlo integration technique can be used, but this introduces extra computations since the samples are not intrinsic in the algorithm. For details see for instance [17], where this idea was tested.

6 Conclusions

A sensor fusion approach to find estimates of the tool position by combining a 3-axes accelerometer and the measurements from the motor angles of an industrial robot is presented. The estimation is formulated as a Bayesian problem and two solutions are proposed; extended Kalman filter and particle filter respectively. The algorithms were tested on data from a realistic robot model. For the linear dynamical model used in the estimation, sufficiently accurate estimates are produced. The performance both with and without accelerometer measurements are close to the fundamental Cramér-Rao lower bound limit. Estimation performance with the accelerometer is better, considering both the Cramér-Rao lower bound and the actual result from the Monte Carlo simulations. A comparison of a filter-based and a non-filter based approach to find estimates of the arm angle shows a significant improvement using the model-based filter. The velocity estimates are also proven to be much more accurate when the filter uses information from the accelerometer. This is important for control design in order to give a well damped response at the robot arm. Under the assumption that the robot can be statically calibrated the estimation algorithms are shown to be very robust to variations in the spring constant and the damping coefficient. Since the intended use of the estimates is to improve position control using an off-line method, like ILC, there are no real-time issues using the computational demanding particle filter algorithm, however the extended Kalman filter runs in real-time in MATLAB. The estimation methods presented in this paper are general and can be extended to higher degrees of freedom robots and additional sensors, such as accelerometers, gyros, or camera systems, can be included. The main effect is a more complex measurement equation.

Acknowledgment

This work was supported by the VINNOVA's Center of Excellence ISIS (Information Systems for Industrial Control and Supervision) at Linköping University, Sweden. Especially the partner ABB.

References

[1] B.D.O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, NJ, 1979.

[2] S. Arimoto, S. Kawamura, and F. Miyazaki. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2):123–140, 1984.

[3] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, February 2002.

[4] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Linköping Studies in Science and Technology. Dissertations No. 579, Linköping University, Linköping, Sweden, 1999.

[5] P. I. Corke. A robotics toolbox for MATLAB. *IEEE Robotics and Automation Magazine*, 3(1):24–32, March 1996.

[6] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[7] N. J. Gordon, D. J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[8] S. Gunnarsson and M. Norrlöf. Iterative learning control of a flexible robot arm using accelerometers. In *IEEE Conference on Control Applications*, Taipei, Taiwan, September 2004.

[9] S. Gunnarsson, M. Norrlöf, G. Hovland, U. Carlsson, T. Brogårdh, T. Svensson, and S. Moberg. Pathcorrection for an industrial robot. European Patent Application No. EP1274546, April 2001.

[10] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons Ltd, 2000.

[11] R. Jassemi-Zargani and D. Necsulescu. Extended Kalman filter-based sensor fusion for operational space control of a robot arm. *IEEE Transactions on Instrumentation and Measurement*, 51(6):1279 – 1282, December 2002.

[12] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, 1970.

[13] P. Jensfelt. *Approaches to Mobile Robot Localization in Indoor Environments*. PhD thesis, Royal Institute of Technology, 2001.

[14] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Information and System Sciences. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[15] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the AMSE–Journal of Basic Engineering*, 82:35–45, 1960.

[16] R. Karlsson. *Simulation Based Methods for Target Tracking*. Linköping Studies in Science and Technology. Licentiate Thesis No. 930, Linköping University, Linköping, Sweden, February 2002.

[17] R. Karlsson, J. Jansson, and F. Gustafsson. Model-based statistical tracking and decision making for collision avoidance application. In *Proceedings of American Control Conference*, Boston, MA, USA, June 2004.

[18] R. Karlsson and M. Norrlöf. Bayesian position estimation of an industrial robot using multiple sensors. In *Proceedings of the IEEE Conference on Control Applications*, Taipei, Taiwan, September 2004.

[19] S. Kay. *Fundamentals of Statistical Signal Processing*. Prentice Hall, 1993.

[20] K. Kozlowski. *Modelling and Identification in Robotics*. Springer-Verlag, 1998.

[21] C. Kwok, D. Fox, and M. Meila. Real-time particle filters. *Proceedings of the IEEE*, 92(3):469–484, March 2004.

[22] A. D. Luca, G. Paesano, and G. Ulivi. A frequency-domain approach to learning control: Implementation for a robot manipulator. *IEEE Transactions on Industrial Electronics*, 39(1), February 1992.

[23] MathWorks, The MathWorks,Mail 3 Apple Hill Dr. Natick, MA 01760-2098, USA. *SIMULINK: Model-Based and System-Based Design*, 2003.

[24] K. L. Moore. *Iterative Learning Control for Deterministic Systems*. Advances in Industrial Control. Springer-Verlag, London, 1993.

[25] M. Norrlöf. *Iterative Learing Control. Analysis, design and experiments*. PhD thesis, Linköping University, Linköping, Sweden, 2000. Linköping Studies in Science and Technology. Dissertations No. 653.

[26] M. Norrlöf. An adaptive iterative learning control algorithm with experiments on an industrial robot. *IEEE Transactions on Robotics and Automation*, 18(2):245–251, April 2002.

[27] M. Nyström and M. Norrlöf. PGT - A path generation toolbox for Matlab (v0.1). Technical Report LiTH-ISY-R-2542, Department of Electrical Engineering, SE-581 83 Linköping, Sweden, September 2003.

[28] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[29] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Springer, 2000.

[30] H. W. Sorenson. *Recursive Estimation for Nonlinear Dynamic Systems*. In Bayesian Analysis of Time Series and Dynamic Models (Ed. J. C. Spall), Dekker, 1988.

[31] P. Tichavsky, P. Muravchik, and A. Nehorai. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing*, 46(5):1386–1396, 1998.

[32] H. L. Van Trees. *Detection, Estimation and Modulation Theory*. Wiley, New York, 1968.

[33] E. Wernholt and M. Östring. Modeling and control of a bending backwards industrial robot. Technical Report LiTH-ISY-R-2522, Department of Electrical Engineering, SE-581 83 Linköping, Sweden, May 2003.

Paper E

Recursive Bayesian Estimation – Bearings-Only Applications

Edited version of the paper:

R. Karlsson and F. Gustafsson. Recursive Bayesian estimation – bearings-only applications. *Accepted for publication in IEE Proceedings on Radar, Sonar, and Navigation. Special issue on target tracking: Algorithms and Applications.*

Parts of the paper in:

R. Karlsson and F. Gustafsson. Range estimation using angle-only target tracking with particle filters. In *Proceedings of American Control Conference*, volume 5, pages 3743–3748, Arlington, Virginia, USA, June 2001. Invited paper.

R. Karlsson. *Simulation Based Methods for Target Tracking.* Linköping Studies in Science and Technology. Licentiate Thesis No. 930, Linköping University, Linköping, Sweden, February 2002.

Recursive Bayesian Estimation – Bearings-Only Applications

Rickard Karlsson and Fredrik Gustafsson,

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

Abstract

In this paper recursive Bayesian estimation methods are applied to several angle-only applications. Air-to-Air passive ranging, as well as an Air-to-Sea application with terrain induced constraints, are discussed. The bearings-only problem is also discussed using experimental data from a torpedo, i.e., Sea-to-Sea with a passive sonar sensor. The Bayesian estimation problem is solved using the particle filter method and the marginalized particle filter. For comparison, a filter bank method using range parameterized extended Kalman filters is used. In a simulation study the particle filter outperforms the filter bank method.

Keywords: Bearings-only tracking, Particle filter, Marginalized particle filter, Range parameterized extended Kalman filter, Applications.

1 Introduction

Target tracking using angle-only measurements in azimuth and elevation is a common technique for many applications using radar, sonar or *infrared* (IR) sensor information. Typically, radar and sonar sensors are used in an active mode, transmitting energy. In this mode, range and possibly range rate are available from the sensor. To avoid the risk of being detected by a hostile target, it may be desirable to use the sensors in a passive mode. Hence, only angle measurements from target induced energy is available. There are also true passive sensors, such as the IR sensor. The main idea in passive ranging is to use angle information only to estimate the unknown relative range. By causing the observation platform (aircraft, missile, torpedo, etc.) to perform certain maneuvers, it is possible to gain observability of range, although it is often difficult to execute an optimal maneuver. Therefore, the approach is often to perform the maneuvering sequence in a deterministic way, exciting the system sufficiently to gain range observability. The issue of optimal maneuvers is not discussed here, but can be found in [30, 29].

Passive ranging has been an important research area for many years. The classical method is to use a single *extended Kalman filter* (EKF), [2, 20]. A common problem is that a single linearized filter may easily diverge. There exist several approaches to estimate the range using a single tracking filter. As described in [35, 38, 37], a modified spherical/polar coordinate system is preferred to a traditional Cartesian system. In [1], this is studied and it is investigated how to reduce filter divergence problems by selecting the coordinate system. A survey of bearings-only estimators and comparison of methods is given in [17]. Another approach is to use multiple filters, where each filter is parameterized to a range interval. Using a bank of filters, cf. [33], the performance is enhanced using the *range parameterized extended Kalman filter* (RPEKF). Using sequential Monte Carlo methods, or *particle filters*, [11, 14], a single nonlinear filter can be used for passive ranging, [4, 5, 34]. The unknown range uncertainty is easier addressed and constraints due to terrain or limitations in the system can be incorporated in a natural way, [31, 22, 34]. In [31], littoral tracking, i.e., tracking of targets on land and in sea near the boundary region between them is discussed, for a joint tracking and classification problem.

The paper is organized as follows. In Section 2 a common target tracking model for bearings-only is described. In Section 3 the recursive Bayesian estimation problem is formulated together with several methods and algorithms. The EKF and the RPEKF represent the linearized solution assuming Gaussian noise. The *particle filter* (PF) and the marginalized version thereof (MPF) are sample approximations of the general Bayesian problem. Section 4 consists of three bearings-only applications, Air-to-Air, Air-to-Sea and Sea-to-Sea. Simulations and experimental data are considered.

2 Target Tracking Model

There are several possible tracking models for passive ranging problems. The choice of coordinate system, [1, 35, 38, 37], can be one issue. Also modeling of non-kinematic properties, such as signal intensity [8], could in principle improve the result, but in practice it may be troublesome. In this section a simple tracking model is described. The focus is on a discrete-time linear dynamics so all non-linearities appear in the measurement relation. The described model will then be used in all the applications in Section 4. For an overview of models within this structure or other common tracking models see for instance [16, 28, 26, 27].

2.1 Dynamic Model

Assume that the target and the tracking platform are described by the same dynamics, modeled by the linear state equations

$$x_{t+1}^{\mathrm{tg}} = F_t x_t^{\mathrm{tg}} + G_t w_t, \tag{1a}$$

$$x_{t+1}^{\mathrm{o}} = F_t x_t^{\mathrm{o}} + G_t u_t, \tag{1b}$$

where x^{tg} is the tracked object's (target) state vector and x_t^{o} is the tracking platform's state vector. The unknown target input signal is defined as process noise, w_t, and the known input signal is denoted u_t. Hence,

$$x_{t+1} = x_{t+1}^{\text{tg}} - x_{t+1}^{\text{o}} = F_t x_t + G_t w_t + G_t u_t, \tag{2}$$

where $x_t = x_t^{\text{tg}} - x_t^{\text{o}}$ is the relative state vector. Assume that the state vector consists of Cartesian position, $p_t \in \mathbb{R}^M$, and velocity, $v_t \in \mathbb{R}^M$,

$$x_t = \begin{pmatrix} p_t \\ v_t \end{pmatrix}, \tag{3}$$

with the following discrete-time matrices

$$F_t = \begin{pmatrix} \mathcal{I}_M & T \cdot \mathcal{I}_M \\ \mathcal{O}_M & \mathcal{I}_M \end{pmatrix}, \; G_t = \begin{pmatrix} \frac{T^2}{2}\mathcal{I}_M \\ T \cdot \mathcal{I}_M \end{pmatrix}, \tag{4}$$

where $\mathcal{I}_M$ is an $M \times M$ identity matrix, $\mathcal{O}_M$ is an $M \times M$ null matrix and T is the sample time. In Cartesian coordinates the position is given as $p_t = \begin{pmatrix} X_t & Y_t \end{pmatrix}^T$ or $p_t = \begin{pmatrix} X_t & Y_t & Z_t \end{pmatrix}^T$, and the velocity vector as $v_t = \dot{p}_t$.

In the above model the process noise represents the maneuverability of the unknown target. For long range applications, where the target is assumed not to have detected the tracker, a common model used is to assume a straight path, i.e., a small process noise. For maneuvering targets a higher value of the noise can be used, but this will reduce performance. The models and applications presented in the simulations studies can easily be generalized to more complex models and scenarios. Often it may be necessary to introduce multiple turn models. For instance, the *interacting multiple model* (IMM), [6], can be used. Another possibility is to use a change detector and adjust the estimate or system when a maneuver is detected, [18, 17]. In [4] maneuvering target tracking using particle filters and sub-optimal EKF-based filters is discussed.

2.2 Measurement Relation

For bearings-only applications only the azimuth angle, φ_t is measured. For angle-only applications the azimuth angle and the elevation angle, θ_t are measured. In this section these measurement relations are presented.

Bearings-only: The position and velocity vectors consist of $p_t = \begin{pmatrix} X_t & Y_t \end{pmatrix}^T$, $v_t = \begin{pmatrix} \dot{X}_t & \dot{Y}_t \end{pmatrix}^T$. The measurement relation for the azimuth angle, φ, is given as

$$y_t = h(x_t) + e_t = \varphi_t + e_t = \arctan(Y_t/X_t) + e_t. \tag{5}$$

Angle-only: The position and velocity vector consist of $p_t = \begin{pmatrix} X_t & Y_t & Z_t \end{pmatrix}^T$, $v_t =$

$\begin{pmatrix}\dot{X}_t & \dot{Y}_t & \dot{Z}_t\end{pmatrix}^T$. The measurement relation for the azimuth angle, φ, and elevation angle, θ, is given as

$$y_t = h(x_t) + e_t = \begin{pmatrix}\varphi_t \\ \theta_t\end{pmatrix} + e_t = \begin{pmatrix}\arctan(Y_t/X_t) \\ \arctan(\frac{-Z_t}{\sqrt{X_t^2+Y_t^2}})\end{pmatrix} + e_t. \tag{6}$$

3 Bayesian Estimation

Consider the discrete state-space model

$$x_{t+1} = F_t x_t + G_t u_t + G_t w_t, \tag{7a}$$
$$y_t = h(x_t) + e_t, \tag{7b}$$

with state variables $x_t \in \mathbb{R}^m$, input signal u_t and y_t the measurement at time t. Let $\mathbb{Y}_t = \{y_i\}_{i=0}^t$ and assume that the *probability density function* (pdf) for the process noise, $p_w(w)$, and measurement noise $p_e(e)$ are known. Here, only additive noise is considered, but a more general description is possible. The nonlinear prediction density $p(x_{t+1}|\mathbb{Y}_t)$ and filtering density $p(x_t|\mathbb{Y}_t)$ for the Bayesian inference, [19], is given by

$$p(x_{t+1}|\mathbb{Y}_t) = \int_{\mathbb{R}^m} p(x_{t+1}|x_t)p(x_t|\mathbb{Y}_t)\,dx_t, \tag{8a}$$
$$p(x_t|\mathbb{Y}_t) = \frac{p(y_t|x_t)p(x_t|\mathbb{Y}_{t-1})}{p(y_t|\mathbb{Y}_{t-1})}. \tag{8b}$$

These equations are in general not analytically solvable. However, for the important special case of linear-Gaussian dynamics and linear-Gaussian observations the Kalman filter, [21], provides a finite dimensional solution. For a general nonlinear or non-Gaussian system, approximate methods must be used. Here two different approaches of solving the Bayesian equations are considered; EKF and PF. Also the use of a bank of EKFs, for the RPEKF method, is discussed.

3.1 The Extended Kalman Filter

For many nonlinear problems, the noise assumptions and the nonlinearity are such that a linearized solution assuming Gaussian noise will be a good approximation. This is the idea behind the EKF, [2, 20], where the model is linearized around the previous estimate. The time update and measurement update for the EKF are given by

$$\begin{cases}\hat{x}_{t+1|t} = F_t\hat{x}_{t|t} + G_t u_t + G_t w_t, \\ P_{t+1|t} = F_t P_{t|t} F_t^T + G_t Q_t G_t^T,\end{cases} \tag{9a}$$

$$\begin{cases}\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - h(\hat{x}_{t|t-1})), \\ P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}, \\ K_t = P_{t|t-1}H_t^T(H_t P_{t|t-1} H_t^T + R_t)^{-1},\end{cases} \tag{9b}$$

where

$$H_t^T = \nabla_x h^T(x_t)|_{x_t=\hat{x}_{t|t-1}}. \tag{10}$$

The noise covariances are given as

$$Q_t = \mathrm{Cov}\,(w_t)\,,\ R_t = \mathrm{Cov}\,(e_t)\,. \tag{11}$$

3.2 Range Parameterized Extended Kalman Filter

In this section the *range parameterized extended Kalman filter* (RPEKF) is presented. It consists of a bank of individual EKFs, each tuned to a certain range. The RPEKF method described in [25, 5] consists of a bank of EKFs in Cartesian coordinates, initialized with different range assumptions. In [33], the filter bank is expressed in modified polar coordinates.

For a particular EKF the performance is dependent on the *coefficient of variation*, C_R, [33]. To have comparable performance for each filter, the same C_R value should be used on each interval. Approximatively, this is given as $\sigma^{(i)}/r^{(i)}, i = 1, \ldots, N_F$, where $r^{(i)}$ and $\sigma^{(i)}$ are the range and standard deviation for the different filters. In Figure 1, the range intervals are depicted for a predefined interval $(r_{\min}, r_{\max})$. The intervals and the C_R are given as

$$r^{(i)} = \frac{r_{\min}}{2}(\rho^i + \rho^{i-1}), \quad \rho = \Big(\frac{r_{\max}}{r_{\min}}\Big)^{1/N_F}, \tag{12a}$$

$$C_R = \frac{\sigma^{(i)}}{r^{(i)}} = \frac{2(\rho-1)}{\sqrt{12}(\rho+1)}, \tag{12b}$$

Therefore, the variance for each interval is given as $\sigma^{(i)} = r^{(i)} C_R$.

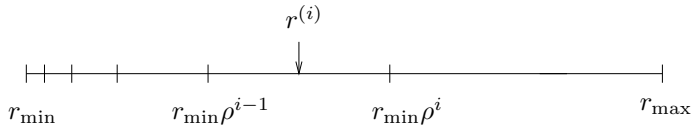


Figure 1: *The RPEKF interval i with mean range $r^{(i)}$.*

The RPEKF uses the likelihood from each EKF, $p(y_t|i)$, to recursively update its probability according to Bayes' rule

$$\gamma_t^{(i)} = p(y_t|i)\gamma_{t-1}^{(i)}. \tag{13}$$

The prior distribution is assumed uniform, i.e., $\gamma_0^{(i)} = \frac{1}{N_F}$, $i = 1, \ldots, N_F$. However, if other information is available it could be used to enhance the performance. The

likelihood is given from the EKF as

$$p(y_t|i) \propto \frac{1}{\sqrt{\det(S_t^{(i)})}} e^{-\frac{1}{2}\left(y_t - h(\hat{x}_{t|t-1}^{(i)})\right)^T S_t^{(i)\,-1}\left(y_t - h(\hat{x}_{t|t-1}^{(i)})\right)}, \tag{14a}$$

$$S_t^{(i)} = H_t^{(i)} P_{t|t-1}^{(i)} (H_t^{(i)})^T + R_t. \tag{14b}$$

The combined estimate and covariance is calculated as

$$\hat{x}_{t|t} = \sum_{i=1}^{N_F} \gamma_t^{(i)} \hat{x}_{t|t}^{(i)}, \tag{15a}$$

$$P_{t|t} = \sum_{i=1}^{N_F} \gamma_t^{(i)} \left(P_{t|t}^{(i)} + (\hat{x}_{t|t}^{(i)} - \hat{x}_{t|t})(\hat{x}_{t|t}^{(i)} - \hat{x}_{t|t})^T \right), \tag{15b}$$

where $P_{t|t}^{(i)}$ is the covariance and $\hat{x}_{t|t}^{(i)}$ the estimate for the different range filter $i = 1, \ldots, N_F$ and the weights $\gamma_t^{(i)}$ are normalized. The RPEKF method is summarized in Algorithm 1.

Algorithm 1 Range Parameterized Extended Kalman Filter (RPEKF)

1: Define $r^{(i)}, \rho, C_R, \sigma^{(i)}$ according to (12a)-(12b), for $i = 1, ..., N_F$.
2: Initialize at $t = 0$: $\hat{x}_{0|0}^{(i)}, P_{0|0}^{(i)}$, $i = 1, ..., N_F$.
3: Given y_t calculate the likelihood using (14).
4: Update the estimate $\hat{x}_{t|t}$ and covariance $P_{t|t}$ according to (15).
5: Predict each EKF in the filter bank, $\hat{x}_{t+1|t}^{(i)}, P_{t+1|t}^{(i)}$ for $i = 1, ..., N_F$ using (9a).
6: Increase t and continue to step 3.

Remark 1: If the filter probability is below a predefined threshold or if some other criterion, such as if the estimated range in a filter is outside the $(r_{\min}, r_{\max})$ interval, the filter is removed from further calculations.
Remark 2: There are many possible ways to initialize a passive ranging estimator. Classical initialization methods are based on measurement initialization for Kalman filters, see for instance [8, 6]. In the implemented angle-only applications, the filter initialization is performed in Cartesian coordinates, projecting the assumed range hypothesis to the *line-of-sight* (LOS) using the measured angles. The velocity consists of the known velocity for the tracking platform. The unknown target velocity is accounted for in the initial uncertainty covariance.

The inital angle measurement is given by

$$y_0 = \begin{pmatrix} \varphi_0 \\ \theta_0 \end{pmatrix}. \tag{16}$$

The initial value of the relative state vector, assuming no knowledge of the target velocity, is for each filter

$$x_0^{(i)} = \begin{pmatrix} r^{(i)} \cos\varphi_0 \cos\theta_0 \\ r^{(i)} \sin\varphi_0 \cos\theta_0 \\ -r^{(i)} \sin\theta_0 \\ 0 - \dot{X}^{\mathrm{o}} \\ 0 - \dot{Y}^{\mathrm{o}} \\ 0 - \dot{Z}^{\mathrm{o}} \end{pmatrix}, \tag{17}$$

where φ_0 and θ_0 are the measured angle values. The initial state covariance matrix in the LOS is given by

$$P_{0_{\mathrm{LOS}}}^{(i)} = \left(\begin{array}{ccc|c} (\sigma^{(i)})^2 & 0 & 0 & \\ 0 & (r^{(i)}\sigma_\varphi)^2 & 0 & \mathcal{O}_3 \\ 0 & 0 & (r^{(i)}\sigma_\theta)^2 & \\ \hline & \mathcal{O}_3 & & \sigma_v^2 \mathcal{I}_3 \end{array}\right),$$

where σ_v is the maximal uncertainty in the target velocity, σ_φ and σ_θ are the angle measurement noise standard deviation and $\sigma^{(i)}$ is the interval uncertainty from (12b). The initial covariance matrix, $P_0^{(i)}$, is calculated as

$$P_0^{(i)} = R_{\mathrm{rot}} P_{0_{\mathrm{LOS}}}^{(i)} R_{\mathrm{rot}}^T,$$

where

$$R_{\mathrm{rot}} = \left(\begin{array}{c|c} (R_{I,B})^T & \mathcal{O}_3 \\ \hline \mathcal{O}_3 & R_{I,B}^T \end{array}\right),$$

$$R_{I,B} = \begin{pmatrix} \cos\varphi_0 & \sin\varphi_0 & 0 \\ -\sin\varphi_0 & \cos\varphi_0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_0 & 0 & -\sin\theta_0 \\ 0 & 1 & 0 \\ \sin\theta_0 & 0 & \cos\theta_0 \end{pmatrix}.$$

For the case when only the azimuth angle is measured, a similar initialization method is used.

3.3 The Particle Filter

In this section the *particle filter* (PF) theory is presented according to [7, 11, 14, 22, 34]. The particle filter provides an approximative solution to the discrete time Bayesian estimation problem formulated in (8) by updating an approximate description of the posterior filtering density. The particle filter approximates the density $p(x_t|\mathbb{Y}_t)$ by a large set of N samples (particles), $\{x_t^{(i)}\}_{i=1}^N$, where each particle has an assigned relative weight, $\gamma_t^{(i)}$, chosen so that all weights sum to unity. The location and weight of each particle reflect the value of the density in that region of state space. The particle filter updates the particle location and the

corresponding weights recursively with each new observed measurement. Since the measurement noise is assumed additive, the unnormalized weights are given by

$$\gamma_t^{(i)} = p_e(y_t - h(x_t^{(i)})), \quad i = 1, \ldots, N. \tag{18}$$

Using the samples (particles) and the corresponding weights the Bayesian equations can be approximately solved. To avoid divergence a resampling step is introduced. This is referred to as the *Sampling Importance Resampling* (SIR), [14], and is summarized in Algorithm 2.

Algorithm 2 Sampling Importance Resampling (SIR)

1: Let $t = 0$. Generate N samples $\{x_0^{(i)}\}_{i=1}^N$ from $p(x_0)$.
2: Compute $\gamma_t^{(i)} = p_e(y_t - h(x_{t|t-1}^{(i)}))$ and normalize, i.e., $\bar{\gamma}_t^{(i)} = \gamma_t^{(i)} / \sum_{j=1}^N \gamma_t^{(j)}$, $i = 1, \ldots, N$.
3: Resample N particles with replacement according to,

$$\text{Prob}(x_{t|t}^{p,(i)} = x_{t|t-1}^{p,(j)}) = \tilde{\gamma}_t^{(j)}. \tag{19}$$

4: $x_{t+1|t}^{(i)} = F_t x_{t|t}^{(i)} + G_t u_t + G_t w_t^{(i)}$, $i = 1, \ldots, N$ using different noise realizations, $w_t^{(i)}$.
5: Increase t and iterate to step 2.

As the estimate for each time, choose the minimum mean square estimate, i.e.,

$$\hat{x}_t = \mathbb{E}\,(x_t|\mathbb{Y}_t) = \int_{\mathbb{R}^m} x_t p(x_t|\mathbb{Y}_t) dx_t \approx \sum_{i=1}^N \gamma_t^{(i)} x_t^{(i)}. \tag{20}$$

The particle filter approximates the posterior pdf, $p(x_t|\mathbb{Y}_t)$, by a finite number of particles. However, asymptotically the approximated pdf converges to the true one, [11].

Remark 1: The particle filter for bearings-only estimation is easily initialized assuming a uniform distribution in the unknown range in the direction of the observed angles.

The terrain constraints can be formulated as a nonlinear state update equation. The terrain map could also be interpreted as a sensor, which would affect the likelihood function. Denote the set of allowed positions, $\Omega = \{X, Y : (X, Y) \in \text{Sea region}\}$.

3.3.1 Terrain Constraints via Time Update

The terrain constraints are in this case handled in the update equation, where particles are accepted if the predicted position, after the additive process noise has

been added, belongs to an accaptable region, Ω, in the database.

$$p_{w_t}(w_t) \propto \begin{cases} p_{\bar{w}_t}(\bar{w}), & x_{t+1}^{\mathrm{tg},(i)} \in \Omega, \\ 0, & x_{t+1}^{\mathrm{tg},(i)} \notin \Omega. \end{cases}$$

3.3.2 Terrain Constraints via Measurement Update

A natural approach to introduce constraints is by using the importance weights calculated in the measurement update. This basically means that the database acts as an extra sensor.

$$p_{e_t}(e_t) \propto \begin{cases} p_{\bar{e}_t}(y_t - h(x_t^{(i)})), & x_t^{\mathrm{tg},(i)} \in \Omega \\ 0, & x_t^{\mathrm{tg},(i)} \notin \Omega. \end{cases}$$

It is straightforward to impose other constraints. For example on state-variables such as velocity and acceleration. The fact that sea-targets are close to the surface is also handled. All these constraints are difficult or troublesome to fulfill with classical Kalman filter techniques.

3.4 The Marginalized Particle Filter

If the state-space model contains linear-Gaussian substructures this can be used to obtain better estimates using the *marginalized particle filter* (MPF), also denoted the Rao-Blackwellized particle filter, [12, 9, 13, 10, 3, 13, 36, 32, 16]. For many target tracking applications it is common to use linear-Gaussian dynamics. All nonlinearities are then in the measurement relation. Particularly, if not all state variables are present in the measurement relation marginalization can be applied. This is the case for all the applications described in this paper.

Consider the following partition of the system

$$x_{t+1}^p = A_t^p x_t^p + A_t^k x_t^k + w_t^p, \qquad w_t^p \in \mathcal{N}(0, Q_t^p), \tag{21a}$$

$$x_{t+1}^k = F_t^p x_t^p + F_t^k x_t^k + w_t^k, \qquad w_t^k \in \mathcal{N}(0, Q_t^k), \tag{21b}$$

$$y_t = h_t(x_t^p) + e_t, \qquad e_t \in \mathcal{N}(0, R_t), \tag{21c}$$

where the noise is assumed to be independent.

Denote the state vector $x_t = \begin{pmatrix} x_t^p & x_t^k \end{pmatrix}^T$, with linear states x_t^k and nonlinear states x_t^p. Furthermore, $\mathbb{X}_t^p = \{x_i^p\}_{i=0}^t$ and $\mathbb{Y}_t = \{y_i\}_{i=0}^t$. Using

$$p(\mathbb{X}_t^p, x_t^k | \mathbb{Y}_t) = p(x_t^k | \mathbb{X}_t^p, \mathbb{Y}_t) p(\mathbb{X}_t^p | \mathbb{Y}_t), \tag{22}$$

where $p(\mathbb{X}_t^p | \mathbb{Y}_t)$ is given by a PF and $x_t^k | \mathbb{X}_t^p$ is linear-Gaussian, i.e., $p(x_t^k | \mathbb{X}_t^p, \mathbb{Y}_t)$ is given by the Kalman filter.

The assumption of independent noise is not necessary. If the noise has dependencies, which most tracking system have, these can be handled by decorrelating the noises using a Gram-Schmidt procedure according to what has been done in [20].

Algorithm 3 Marginalized Particle Filter (MPF)

1: Initialization: For $i = 1, \ldots, N$, initialize the particles, $x_{0|-1}^{p,(i)} \sim p_{x_0^p}(x_0^p)$ and set $\{x_{0|-1}^{k,(i)}, P_{0|-1}^{(i)}\} = \{\bar{x}_0^k, \bar{P}_0\}$. Set $t = 0$.

2: For $i = 1, \ldots, N$, evaluate the importance weights $\gamma_t^{(i)} = p(y_t|\mathbb{X}_t^{p,(i)}, \mathbb{Y}_{t-1})$ according to the likelihood

$$p(y_t|\mathbb{X}_t^p, \mathbb{Y}_{t-1}) = p_e(y_t - h_t(x_t^p)), \tag{23}$$

and normalize $\tilde{\gamma}_t^{(i)} = \frac{\gamma_t^{(i)}}{\sum_{j=1}^N \gamma_t^{(j)}}$.

3: PF measurement update: Resample N particles with replacement according to,

$$\text{Prob}(x_{t|t}^{p,(i)} = x_{t|t-1}^{p,(j)}) = \tilde{\gamma}_t^{(j)}. \tag{24}$$

4: PF time update and Kalman filter update

(a) Kalman filter measurement update,

$$\hat{x}_{t|t}^{k,(i)} = \hat{x}_{t|t-1}^{k,(i)},\ P_{t|t} = P_{t|t-1}. \tag{25}$$

(b) PF time update: For $i = 1, \ldots, N$,

$$x_{t+1|t}^{p,(i)} \sim p(x_{t+1|t}^p|\mathbb{X}_t^{p,(i)}, \mathbb{Y}_t), \tag{26}$$

where

$$p(x_{t+1|t}^{p,(i)}|\mathbb{X}_t^{p,(i)}, \mathbb{Y}_t) = \mathcal{N}(A_t x_t^{p,(i)} + A_t^k \hat{x}_{t|t}^{k,(i)},\ A_t^k P_{t|t}(A_t^k)^T + Q_t^p).$$

(c) Kalman filter time update,

$$\begin{aligned}
\hat{x}_{t+1|t}^{k,(i)} &= F_t^k \hat{x}_{t|t}^{k,(i)} + F_t^p x_t^{p,(i)} + L_t(x_{t+1|t}^{p,(i)} - A_t^p x_t^{p,(i)} - A_t^k \hat{x}_{t|t}^{k,(i)}),\\
P_{t+1|t} &= F_t^k P_{t|t}(F_t^k)^T + Q_t^k - L_t M_t L_t^T,\\
M_t &= A_t^k P_{t|t}(A_t^k)^T + Q_t^p,\\
L_t &= F_t^k P_{t|t}(A_t^k)^T M_t^{-1},
\end{aligned}$$

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

In Algorithm 3 the MPF is summarized for the model given in (21) with uncorrelated process noise, in order to keep the algorithm simple.

Remark 1: If the tracking model from Section 2 is used in the MPF, relevant matrices must be re-defined. Consider $\bar{w}_t = Gw_t$. Hence

$$\bar{Q}_t = \mathbb{E}\left(\bar{w}_t \bar{w}_t^T\right) = GQ_tG^T = \begin{pmatrix} Q_t^p & \mathcal{S}_t \\ \mathcal{S}_t^T & Q_t^k \end{pmatrix}. \tag{28}$$

In order to adjust for the correlated process noise, define the scalar product of two stochastic signals as $<u, v> = \mathbb{E}\left(uv^T\right)$. According to [20], the noise can be decorrelated by

$$\begin{aligned} \tilde{w}_t^k &= w_t^k - <w_t^k, w_t^p> \|w_t^p\|^{-2} w_t^p \\ &= w_t^k - \mathcal{S}_t (Q_t^p)^{-1} w_t^p. \end{aligned} \tag{29}$$

where $<w_t^p, w_t^k> = \mathcal{S}_t$. Once step (4b) in Algorithm 3 has been executed w_t^p is known and it is straightforward to calculate $\tilde{w}_t^k$, and the decorrelation is complete.

Remark 2: In the model only the position states, X_t, Y_t and Z_t, are present in the measurement relation. Using the notation from Section 3.4 this means that $x_t^p = \begin{pmatrix} X_t & Y_t & Z_t \end{pmatrix}^T$ and $x_t^k = \begin{pmatrix} \dot{X}_t & \dot{Y}_t & \dot{Z}_t \end{pmatrix}$. Hence, the particle filter dimension is reduced from $\mathbb{R}^6$ to $\mathbb{R}^3$. Even though the MPF introduces extra calculations this is a more efficient method for many system, since the number of particles needed can be reduced. In [24] this is analyzed more thoroughly, where the computational complexity of the MPF is discussed.

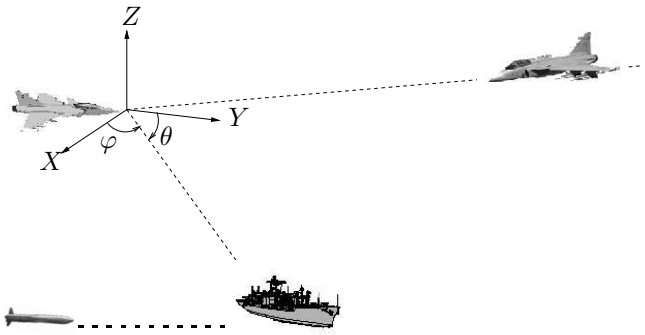


Figure 2: *Passive ranging for Air-to-Air, Air-to-Sea, and Sea-to-Sea applications. The relative azimuth (φ) and elevation (θ) angles are defined from the tracking platform.*

4 Passive Ranging Applications

In this section several passive ranging applications are presented using the estimation methods described in Section 3. In Figure 2 the Air-to-Air and Air-to-Sea applications are depicted. Monte Carlo evaluations on simulated data as well as true experimental data for an Sea-to-Sea application are presented. The estimation performance is evaluated in a Monte Carlo simulation study using the same scenario, but with different measurement noise realizations and randomly chosen initial target values. The performance is evaluated using the *root mean square error* (RMSE) for each time according to [15]

$$\mathrm{RMSE}(t) = \left(\frac{1}{N_{\mathrm{MC}}} \sum_{j=1}^{N_{\mathrm{MC}}} \| x_t^{\mathrm{TRUE}} - \hat{x}_t^{(j)} \|_2^2 \right)^{1/2}, \tag{30}$$

where N_{MC} is the number of Monte Carlo simulation, and where $\hat{x}^{(j)}$ denotes the estimate at time t for Monte Carlo simulation j and where x_t^{TRUE} is the true value.

In all applications the RPEKF is initialized as described in Section 3.2 and the particle filter is initialized as described in Section 3.3. The number of particles is very large initially, in order to have an accurate range resolution, and it is reduced deterministically.

4.1 Air-to-Air Passive Ranging

This section is based on part of the passive ranging application from [23], where the RPEKF is compared to the PF method in a Monte Carlo simulation study for an Air-to-Air application. Assume that the target is non-maneuvering and that angle observations are available with a sample period of $T = 1\,\mathrm{s}$. The simulation parameters are presented in Table 1.

Table 1: *Air-to-Air simulation parameters.*

Monte Carlo simulations	$N_{MC} = 100$
Process noise covariance	$Q = \mathrm{diag}(4,\ 4,\ 4)$
Measurement noise	$\sigma_\varphi = \sigma_\theta = 1\,\mathrm{mrad}$ (Gaussian)
Min/Max distances	$R_{\min} = 10\,\mathrm{km},\ R_{\max} = 30\,\mathrm{km}$
Sample time	$T = 1\,\mathrm{s}$
Number of particles	$N = 80000 \searrow 10000$
Number of RPEKFs	$N_F = 6$
Relative height	1000 m
Speed	$v^o = 150,\ v^{\mathrm{tg}} = 150$ m/s
Input signal (maneuver)	$u = \pm 4$ g (lateral)

In Figure 3 the scenario is presented. The position RMSE is given in Figure 4. As seen the PF and MPF have better RMSE performance, but asymptotically the RPEKF performs almost as well.

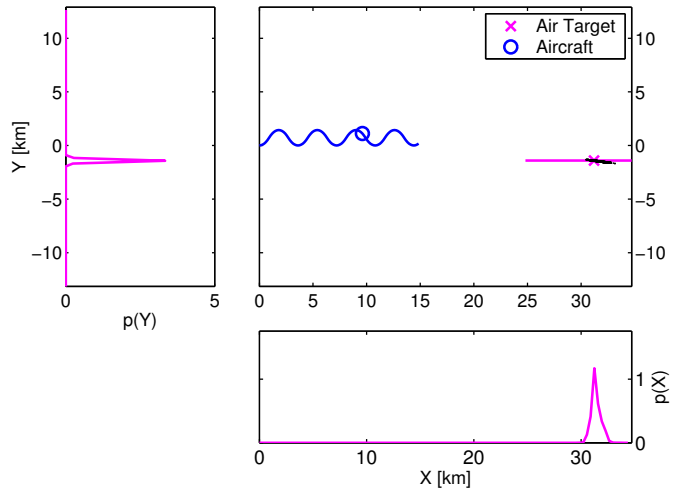


Figure 3: *The scenario in the (X,Y)-plane. The aircraft and the target trajectories are shown. The position at $t = 70$ is indicated together with the particle cloud from one realization of the particle filter. Marginalized position pdfs represent the estimated position uncertainty.*

4.2 Air-to-Sea Passive Ranging

In this section an Air-to-Sea passive ranging application from [22] is presented as illustrated in Figure 2. The main objective in this section is to merge terrain type information from a terrain database to the kinematic part in order to discard regions uninteresting to the tracking application. Terrain induced tracking constraints improve tracking performance and reduce the computational complexity. In this application the filter only distinguishes between land and sea, but for other applications, more detailed terrain type information could be used. In the particle filter, particles are discarded if the position is within a prohibited area (land), whereas particles belonging to feasible regions (sea) are accepted and assigned a probability according to the likelihood function.

In a simulation study the range estimation problem using an IR sensor is considered. The relative distance and the aircraft's trajectory are illustrated in Figure 5. The target model used in the simulations assumes a small constant velocity. The terrain database has a resolution of 50 m. The simulation parameters are presented in Table 2.

In Figure 5 the scenario is presented together with the marginal position densities in each direction, $p(X)$ and $p(Y)$, for time $t = 1$ s, using terrain constraints.

In Figure 6 the position RMSE is presented for the PF and MPF with and without constraints, and for the RPEKF. As seen the incorporation of constraints

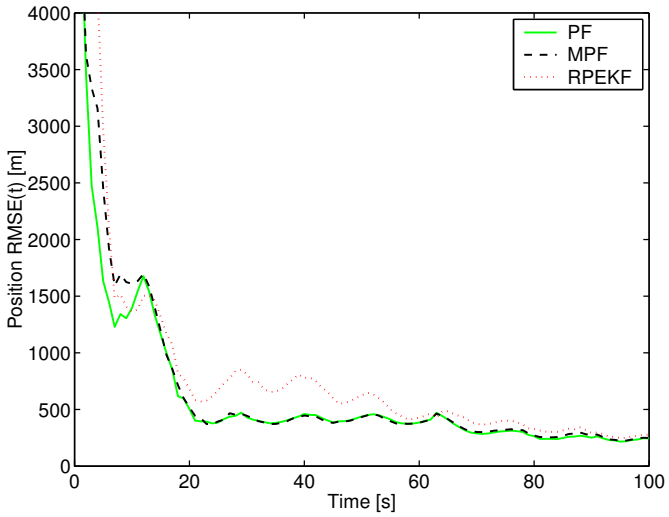


Figure 4: *Position RMSE(t) for PF, MPF, and RPEKF using* 100 *Monte Carlo simulations.*

Table 2: *Air-to-Sea simulation parameters.*

Monte Carlo simulations	$N_{MC} = 100$
Process noise covariance	$Q = \text{diag}(1,\ 1,\ 1)$
Measurement noise	$\sigma_\varphi = \sigma_\theta = 1\,\text{mrad}$ (Gaussian)
Min/Max distances	$R_{\min} = 1\,\text{km},\ \ R_{\max} = 23\,\text{km}$
Sample time	$T = 1\,\text{s}$
Number of particles	$N = 80000 \searrow 10000$
Number of RPEKFs	$N_F = 20$
Relative height	1000 m
Speed	$v^o = 250,\ v^{\text{tg}} \approx 0$ m/s
Input signal (maneuver)	$u = \pm 1$ g (lateral)

improve performance. The different particle filters have basically the same performance for the scenario.

4.3 Sea-to-Sea Passive Ranging

Modern torpedo systems are equipped with an acoustic seeker, which is similar to the electro-magnetic radar. This sound based sensor is referred to as *sonar*. In the active mode range and bearing to a target are available. To avoid being detected by a hostile target and to reduce the risk of hostile counter-measurements, it is often important to minimize the usage of the active mode. In the passive mode,

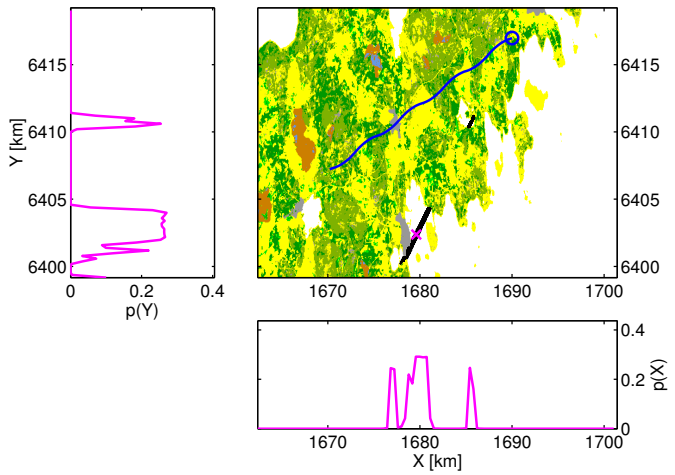


Figure 5: *The sea target position and marginalized position pdf using the particle filter with constraints at* $t = 1\,s$*. The particle cloud and the future trajectory of the aircraft are also shown.*

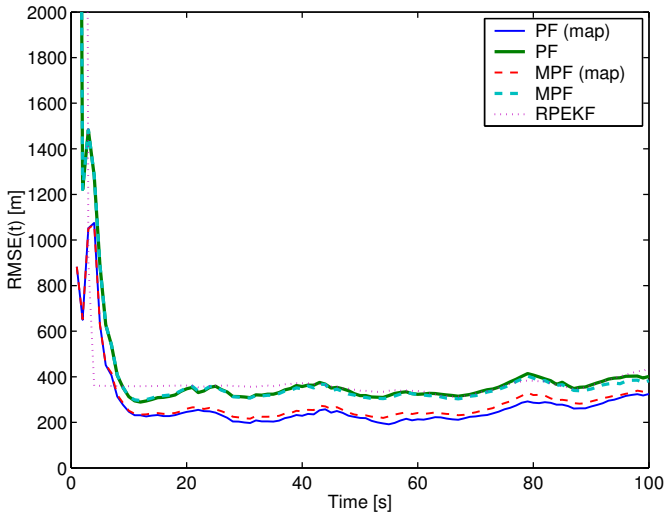


Figure 6: *Position RMSE(t) for Air-to-Sea passive ranging using* 100 *Monte Carlo simulations.*

the acoustic sensor just listens for target related sounds. Hence, in principle only the direction can be measured. Most sonar based systems do not measure the elevation angle, therefore only azimuth bearing information is available.

The bearing information is from experimental sonar data acquired from a torpedo system. The position estimate is done afterward, hence no control feedback to the torpedo. The scenario is given in Figure 7, where torpedo way-points are

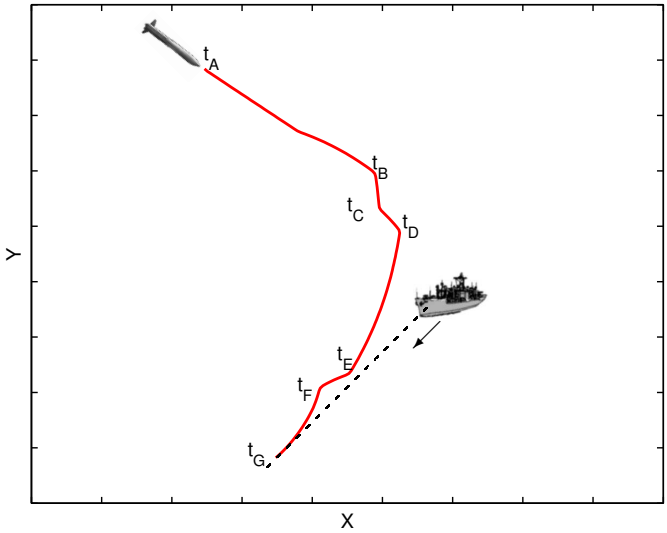


Figure 7: *Torpedo trajectory from experimental data with time indications at way-points and an indication of the ship's true trajectory. The initial time is denoted t_A and the final time t_G.*

marked with time indices. The true position of the target (ship) is not known since no true position information is available. However, it is known that the ship follows a rather straight path, with nearly constant velocity as indicated in the figure. Since the torpedo approaches the target from behind in the final phase ($t > t_F$) and impact occurs at $t = t_G$, the probable approximate true target position is indicated with a dashed line in Figure 7.

Between the initial time t_A and first major maneuver at t_B the torpedo follows a relatively straight trajectory. During this time the range estimation is not particularly good since the small maneuver and the relative geometry to the target does not reveal much range information. At t_B the first major maneuver is performed to gain observability, followed by maneuvers at t_C and t_D. During the long interval $t_D - t_E$ a straight path is followed, where the main objective is to decrease the distance to the ship. Finally, at t_E and t_F maneuvers are performed to gain observability and to approach the target from behind. The total number of measurements is close to 300. The position scale and the actual values of the maneuver times are not presented in any plot, since it is confidential information. The RPEKF and

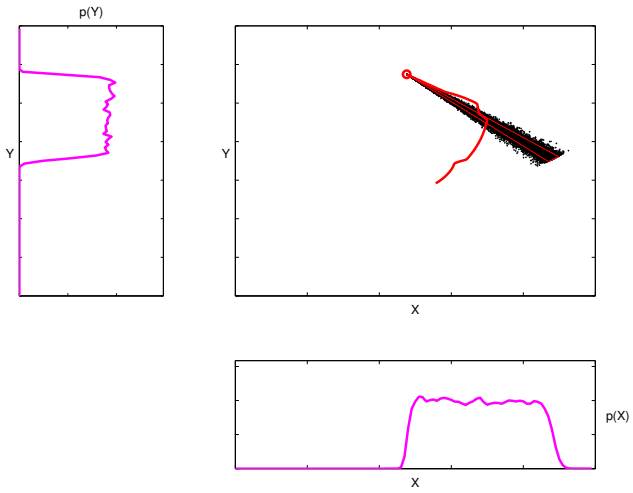


Figure 8: *Torpedo trajectory, particle cloud and marginalized densities for target position at $t = t_A$, in Figure 7.*

the PF are applied to experimental torpedo data, using $N = 15000$ particles and $N_F = 6$ range filters. The number of particles was chosen large enough to minimize effects of to few samples. In Figure 8, the output from the particle filter is shown for $t = t_A$. The target position and the marginalized target position probability densities are given. Initially, no maneuver is made so the range can not be estimated. The full torpedo position trajectory is shown, where the current torpedo position is indicated with a small circle.

Since the relative velocity is estimated, the target speed and heading can be calculated as illustrated in Figure 9 for PF and RPEKF method, respectively. The minimum mean square estimate (20) of target position from the particle filter is presented in Figure 10 together with the estimation from the RPEKF method.

5 Conclusions

In this paper Bayesian estimation methods for several bearings-only applications are discussed. The particle filter can handle any noise distribution. Hence, the initialization is easy and optimal for the bearings-only problem, since it is natural to initialize the filter with a uniform range distribution over all possible target distances. If constraints are present on the system state or from external sources, such as terrain information, the particle filter can easily incorporate these, hence improving performance. This is impossible or troublesome with the EKF approach. The particle filter also improves the estimation performance since no linearizations are necessary. If a linear-Gaussian substructure is present, the MPF can be used,

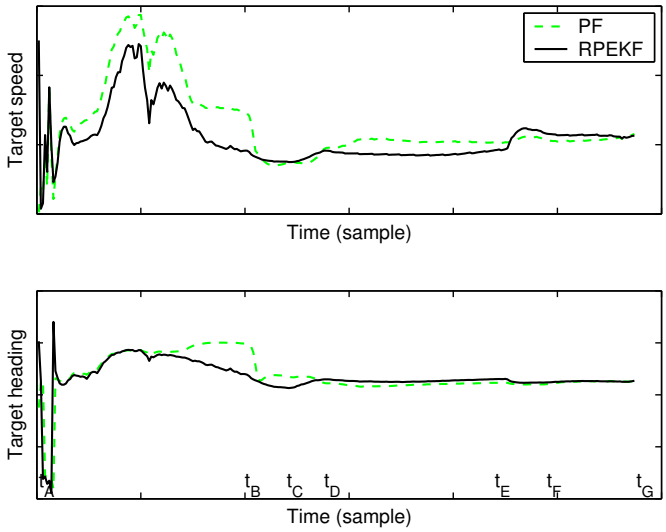


Figure 9: *Target speed and heading estimate, with time indices indicated.*

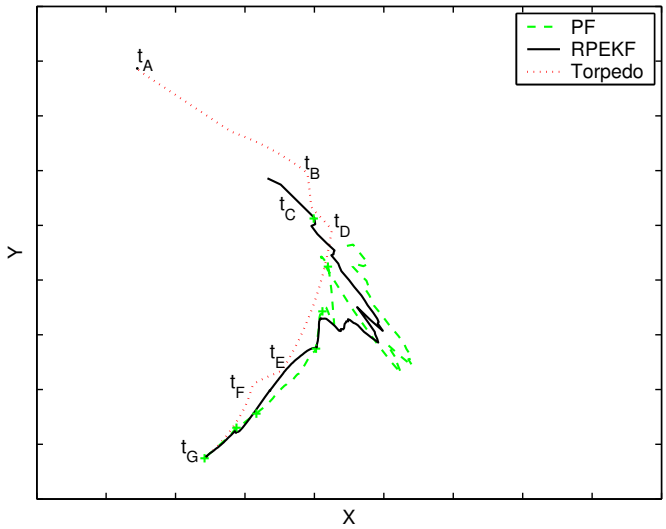


Figure 10: *Torpedo trajectory, minimum mean square estimate of target position using PF and RPEKF methods.*

and possibly to a reduced computational demand. As a comparison to the simulation based particle filter, a bank of EKFs is used, in the RPEKF method. Without terrain constraints, this method yields comparative RMSE results after the initial transient. With constraint present, the particle filters yield better performance. However, the RPEKF method is much faster than the particle filter. Various simulation studies as well as experimental data are presented.

Acknowledgment

The authors would like to thank Saab Bofors Underwater Systems, and especially Per-Ola Svensson and Elias Fransson for providing the sonar data from a torpedo system.

This work was supported by the VINNOVA's Center of Excellence ISIS (Information Systems for Industrial Control and Supervision) at Linköping University, Sweden.

References

[1] V. J. Aidala and S. E. Hammel. Utilization of modified polar coordinates for bearings-only tracking. *IEEE Transactions on Automatic Control*, 28(3):283–294, 1983.

[2] B.D.O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, NJ, 1979.

[3] C. Andrieu and A. Doucet. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society*, 64(4):827–836, 2002.

[4] M. Arulampalam, R. Ristic, N. Gordon, and T. Mansell. Bearings-only tracking of manoeuvring targets using particle filters. *Journal of Applied Signal processing*, 15:2351–2365, 2004.

[5] S. Arulampalam and B. Ristic. Comparison of the particle filter with range-parameterised and modified polar EKFs for angle-only tracking. In *Proceedings of SPIE, Signal and Data Processing of Small Target*, pages 288–299, Orlando, FL, USA, 2000.

[6] Y. Bar-Shalom and X. R. Li. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, 1993.

[7] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Linköping Studies in Science and Technology. Dissertations No. 579, Linköping University, Linköping, Sweden, 1999.

[8] S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[9] G. Casella and C. P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

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

[11] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

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

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

[14] N. J. Gordon, D. J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[15] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons Ltd, 2000.

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

[17] A. Holtsberg. *A Statistical Analysis of Bearings-Only Tracking*. PhD thesis, Department of Mathematical Statistics, Lund University, 1992. LUTFD2/TFMS–1010.

[18] A. Holtsberg and J. Holst. Estimation and confidence in bearings only tracking. In *25th Asilomar Conference on Signals, Systems and Computers*, pages 883–887, 1991.

[19] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, 1970.

[20] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Information and System Sciences. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[21] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the AMSE–Journal of Basic Engineering*, 82:35–45, 1960.

[22] R. Karlsson. *Simulation Based Methods for Target Tracking*. Linköping Studies in Science and Technology. Licentiate Thesis No. 930, Linköping University, Linköping, Sweden, February 2002.

[23] R. Karlsson and F. Gustafsson. Range estimation using angle-only target tracking with particle filters. In *Proceedings of American Control Conference*, volume 5, pages 3743–3748, Arlington, Virginia, USA, June 2001. Invited paper.

[24] R. Karlsson, T. Schön, and F. Gustafsson. Complexity analysis of the marginalized particle filter. Technical Report LiTH-ISY-R-2611, Department of Electrical Engineering, www.control.isy.liu.se/publications, June 2004. To appear in IEEE Transactions on Signal Processing.

[25] T. R. Kronhamn. Bearings-only target motion analysis based on a multihypothesis Kalman filter and adaptive ownship motion control. In *IEE Proceedings on Radar, Sonar and Navigation*, volume 145, pages 247–252, 1998.

[26] X. R. Li and V. P. Jilkov. A survey of maneuvering target tracking–part III: Measurement models. In *Proceedings of SPIE Conference on signal and data processing of small targets*, volume 4473, pages 423–446, July 2001.

[27] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39:1333–1364, October 2003.

[28] X. R. Li and V.P. Jilkov. A survey of maneuvering target tracking: Dynamics models. In *Proceedings of SPIE Conference on signal and data processing of small targets*, volume 4048, pages 212–235, April 2000.

[29] A. Logothetis, A. Isaksson, and R. J. Evans. An information theoretic approach to observer path design for bearings-only tracking. In *Proceedings of the 36:th IEEE Conference on Decision and Control*, volume 4, pages 3132–3137, 1997.

[30] A. Logothetis, A. Isaksson, and R. J. Evans. Comparison of suboptimal strategies for optimal own-ship maneuvers in bearings-only tracking. In *Proceedings of American Control Conference*, volume 6, pages 3334–3338, 1998.

[31] M. Mallick, S. Maskell, T. Kirubarajan, and N. Gordon. Littoral tracking using the particle filter. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 1, pages 935–942, Annapolis, MD, USA, July 2002.

[32] P-J. Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation*. Linköping Studies in Science and Technology. Licentiate Thesis No. 945, Linköping University, Linköping, Sweden, May 2002.

[33] N. Peach. Bearings-only tracking using a set of range-parameterised extended Kalman filters. In *IEE Proceedings of Control Theory and Applications*, volume 142, pages 73–80, January 1995.

[34] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[35] P. N. Robinson and M. R. Yin. Modified spherical coordinates for radar. In *Proceedings AIAA Guidance, Navigation and Control Conference*, pages 55–64, August 1994.

[36] T. Schön, F. Gustafsson, and P-J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *To appear in IEEE Transactions on Signal Processing*.

[37] M. A. Simard and F. Begin. Central level fusion of radar and IRST contacts and the choice of coordinate system. In *SPIE Signal and Data Processing of Small Targets*, volume 1954, pages 462–472, Orlando, FL, USA, July 1993.

[38] D. V. Stallard. An angle-only tracking filter in modified sperical coordinates. In *Proceedings AIAA Guidance and Navigation and Control Conference*, pages 542–550, 1987.

Paper F

Monte Carlo Data Association for Multiple Target Tracking

Edited version of the paper:

R. Karlsson and F. Gustafsson. Monte Carlo data association for multiple target tracking. In *IEE International Seminar on Target Tracking: Algorithms and Applications*, Enschede, The Netherlands, October 2001.

Monte Carlo Data Association for Multiple Target Tracking

Rickard Karlsson and Fredrik Gustafsson,

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

Abstract

The data association problem occurs for multiple target tracking applications. Since nonlinear and non-Gaussian estimation problems are solved approximately in an optimal way using recursive Monte Carlo methods or particle filters, the association step will be crucial for the overall performance. We introduce a Bayesian data association method based on the particle filter idea and the joint probabilistic data association (JPDA) hypothesis calculations. A comparison with classical EKF based data association methods such as the nearest neighbor (NN) method and the JPDA method is made. The NN association method is also applied to the particle filter method. Multiple target tracking using particle filter will increase the computational burden, therefore a control structure for the number of samples needed is proposed. A radar target tracking application is used in a simulation study for evaluation.

Keywords: Multiple target tracking, Data association, Particle filter.

1 Introduction

For multiple target tracking application the data association problem must be handled. Traditionally, the estimation problem is solved using linearized filters, such as the *extended Kalman filter* (EKF) [1], under a Gaussian noise assumption. The sufficient statistics from the linearized filter are used for data association. Several classical association methods have been proposed in the literature. When dealing with nonlinear models in state equation and measurement relation and a non-Gaussian noise assumption, these estimation methods may lead to non-optimal solutions. The sequential Monte Carlo methods, or particle filters, provide general solutions to many problems where linearizations and Gaussian approximations are

intractable or would yield too low performance. In this paper, we apply the classical particle filter *Bayesian bootstrap* [12], to a multiple target environment. In a simulation study we compare this approach to traditional methods. To handle the complexity problem we also propose a controller structure, to recursively chose the number of particles.

2 Sequential Monte Carlo Methods

Monte Carlo techniques have been a growing research area lately due to improved computer performance. A rebirth of this type of algorithms came after the seminal paper of Gordon et al. [12], showing that Monte Carlo methods could be used in practice to solve the optimal estimation problem. In the recent article collection, [9], the theory and development in sequential Monte Carlo methods over the last years are summarized.

Consider the following nonlinear discrete time system for a single target

$$\begin{aligned} x_{t+1} &= f(x_t) + w_t, \\ y_t &= h(x_t) + e_t. \end{aligned}$$

The sequential Monte Carlo methods, or particle filters, provide an approximative Bayesian solution to discrete time recursive problem by updating an approximative description of the posterior filtering density. Let $x_t \in \mathbb{R}^n$ denote the state of the observed system and $\mathbb{Y}_t = \{y_i\}_{i=0}^t$ be the set of observations until present time. Assume independent process noise w_t and measurement noise e_t with densities p_{w_t} respective p_{e_t}. The initial uncertainty is described by the density p_{x_0}. The particle filter approximates the probability density $p(x_t|\mathbb{Y}_t)$ by a large set of N particles $\{x_t^{(i)}\}_{i=1}^N$, where each particle has an assigned relative weight, $\gamma_t^{(i)}$, such that all weights sum to unity. The location and weight of each particle reflect the value of the density in the region of the state space. The particle filter updates the particle location and the corresponding weights recursively with each new observation. The nonlinear prediction density $p(x_t|\mathbb{Y}_{t-1})$ and filtering density $p(x_t|\mathbb{Y}_t)$ for the Bayesian inference are given by

$$p(x_t|\mathbb{Y}_{t-1}) = \int_{\mathbb{R}^n} p(x_t|x_{t-1}) p(x_{t-1}|\mathbb{Y}_{t-1}) dx_{t-1}, \tag{1}$$

$$p(x_t|\mathbb{Y}_t) \propto p(y_t|x_t) p(x_t|\mathbb{Y}_{t-1}). \tag{2}$$

The main idea is to approximate $p(x_t|\mathbb{Y}_{t-1})$ with

$$p(x_t|\mathbb{Y}_{t-1}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta(x_t - x_t^{(i)}), \tag{3}$$

where δ is the discrete Dirac function. Inserting (3) into (2) yields a density to sample from. This can be done by using the Bayesian bootstrap or *sampling importance resampling* (SIR) algorithm from [12], given in Algorithm 1. The estimate

Algorithm 1 Sampling Importance Resampling (SIR).

1: Set $t = 0$, generate N samples $\{x_0^{(i)}\}_{i=1}^N$ from the initial distribution $p(x_0)$.
2: Compute the weights $\gamma_t^{(i)} = p(y_t|x_t^{(i)})$ and normalize, i.e., $\tilde{\gamma}_t^{(i)} = \gamma_t^{(i)} / \sum_{j=1}^N \gamma_t^{(j)}, i = 1, \ldots, N$.
3: Generate a new set $\{x_t^{(i\star)}\}_{i=1}^N$ by resampling with replacement N times from $\{x_t^{(i)}\}_{i=1}^N$, where $\text{Prob}(x_t^{(i\star)} = x_t^{(j)}) = \tilde{\gamma}_t^{(j)}$.
4: Predict (simulate) new particles, i.e., $x_{t+1}^{(i)} = f(x_t^{(i\star)}, w_t^{(i)}), i = 1, \ldots, N$ using different noise realizations for the particles.
5: Increase t and iterate to item 2.

and uncertainty region for the particle filter can be calculated as

$$\hat{x}_{t|t}^{\text{MMS}} = \sum_{i=1}^{N} \gamma_t^{(i)} x_t^{(i)}, \tag{4}$$

$$P_{t|t} = \sum_{i=1}^{N} \gamma_t^{(i)} (x_t^{(i)} - \hat{x}_{t|t}^{\text{MMS}})(x_t^{(i)} - \hat{x}_{t|t}^{\text{MMS}})^T. \tag{5}$$

3 Particle Number Controller

The computational burden for the particle filter is dependent on the number of particles and on the resampling calculation. However, the resampling can be efficiently implemented using a classical algorithm for sampling N ordered independent identically distributed variables [5, 17]. For multiple target tracking applications the computational burden is increased. Therefore, it is essential to minimize the number of particles used in the estimation step. A novel approach is to apply a simple control structure according to Figure 1. The number of particles needed is deter-

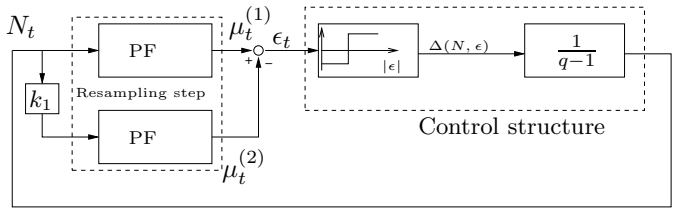


Figure 1: *Controller of particles.*

mined by the controller using the residual $\epsilon_t = \|\mu_t^{(1)} - \mu_t^{(2)}\|$, where $\mu_t^{(1)}$ and $\mu_t^{(2)}$ are some statistical property from the particle filters (PFs), using different number of particles. Possible choices are for instance some relevant statistics, such as the mean estimate from the particle filter or utilization of the probability density

(pdf) or the cumulative density function (cdf). For instance the marginal distribution (density for each coordinate) could be used. The control structure used is a nonlinear block consisting of a relay and an integrator using

$$\Delta(N_t, \epsilon_t) = \begin{cases} \alpha_{\text{inc}}(N_t) & \text{, if } |\epsilon_t| > \Lambda \\ \alpha_{\text{dec}}(N_t) & \text{, if } |\epsilon_t| \leq \Lambda \end{cases},$$

For maneuvering targets in a tracking application the controller can reduce or increase the number of particles during the tracking envelope. However, performance may now depend on the parameters of the controller. Note that the controller is implemented in the resampling step (Algorithm 1, step 3).

4 Data Association

Data association is a problem of great importance for multiple target tracking applications. Several methods have been proposed in the literature and different methods are often discussed in estimation and tracking literature, [3, 4, 7, 8]. In general multi target tracking deals with state estimation of an unknown number of targets. Some methods are special cases which assume that the number of targets is constant or known. The observations are considered to originate from targets if detected or from clutter. The clutter is a special model for false alarms, whose statistical properties are different from the targets. In some applications only one measurement is assumed from each target object, where in other applications several returns are available. This will of course reflect which data association method to use.

Several classical data association methods exist. The simplest is probably the *nearest neighbor* (NN). In [3], this is referred to as the *nearest neighbor standard filter* (NNSF) and uses only the closest observation to any given state to perform the measurement update step. The method can also be given as a global optimization, so the total observation to track statistical distance is minimized. Another multi target tracking association method is the *joint probability data association* (JPDA) which is an extension of the *probability data association* (PDA) algorithm to multi targets. It estimates the states by a sum over all the association hypothesis weighted by the probabilities from the likelihood. The most general method is a time-consuming algorithm called the *multi hypothesis tracking* (MHT), which calculates every possible update hypothesis. In [16], several algorithms for multiple target tracking are listed and categorized according to the underlying assumptions. A reference list to the different methods is also given. In [15], the *probabilistic* MHT (PMHT) method is presented, using a maximum-likelihood method in combination with the *expectation maximization* (EM) method. A comparison between the JPDAF and the PMHT is also made. In [6], a *Markov Chain Monte Carlo* (MCMC) technique is used for data association of multiple measurements in an over the horizon radar application.

Most of these methods rely upon that the mean and covariance is sufficient statistics for the problem. For linear and Gaussian problems the Kalman filter

is the optimal estimator yielding sufficient information. For nonlinear problems the EKF is often used as an approximation. To be able to fully use nonlinear and non-Gaussian estimation methods combined with data association to solve the joint data association and estimation problem there is a need to develop other methods. In [2], the solution to the assignment problem for data association is proposed to be within the Bayesian framework by simply incorporate it in the estimation equations. In [18], this idea is suggested for the particle filter, when the problem of maintaining a track on a target in the presence of intermittent spurious objects. In [11], a multiple target and multiple sensor estimation and association problem is solved using the Bayesian bootstrap filter. Samples are drawn from the overall target probability density. A special filter called *hybrid bootstrap filter* is constructed. The *joint-filter* in [14], is a solution to the joint data association and estimation problem for particle filters. The estimation is done using a particle filter and a Gibbs sampler, [10], is used for the association. The case for unknown number of targets is handled by using a hypothesis test.

In this paper we focus on this idea for a multiple target problem in a cluttered environment, and compare the particle filter based estimation and association with classical association techniques.

5 Monte Carlo Probabilistic Data Association

In this paper we modify the classical SIR algorithm (Algorithm 1) for estimation to handle multiple targets. The association principle proposed is based on a novel Monte Carlo approach for the JPDA algorithm. We have assumed time-invariant target models, which are the same for all targets. We use the same Bayesian approach as in [11], for the estimation. However, we extend the idea and introduce hypothesis calculations according to the JPDA method. The resampling is then executed over all target association hypotheses. The clutter or false alarm model is assumed uniformly distributed in the volume and the number of false alarms for a given time is assumed to be Poisson distributed.

Let x_t be the state at time t for the relative target locations, i.e., $x_t = \{x_t^1, \ldots, x_t^\tau\}$. The samples or particles in the SIR/MCJPDA method is defined as $\{x_t^{(i)}\}_{i=1}^{N_t} = \{x_0^{(i),1}, \ldots, x_0^{(i),\tau}\}_{i=1}^{N_t}$, where each initial target cloud is denoted $x_0^{(i),j}$ for targets $j = 1, \ldots, \tau$. The measurements for each time frame (scan) are denoted y_t^k, $k = 1, \ldots, M_t$. A special clutter model is used to handle false alarms, x_t^0 $(j = 0)$. The association likelihood (track j, measurement k) is given by $p_{jk} = p_{e_t}(y_t^k - h(x_t^j))$. A general expression for the probability in hypothesis H_n is:

$$\text{Prob}(H_n) = \delta_n P_D^{\tau - Z_n} (1 - P_D)^{Z_n} P_{FA}^{M_t - (\tau - Z_n)} l_n, \tag{6}$$

where Z_n is the number of false alarms (FA) in hypothesis n and l_n is the likelihood part. For more details, see hypothesis calculations in the example given in [8] (p.

354). We also have an extra option

$$\delta_n = \begin{cases} 1, \text{allow multiple measurement associations} \\ 0, \text{otherwise.} \end{cases}$$

For the particle filter each particle is associated with a weight:

$$\gamma_t^{(i)} = \sum_{n=1}^{(M_t+1)^\tau} \text{Prob}(H_n^{(i)}).$$

Normalization yields the particle probability $\tilde{\gamma}_t^{(i)}$. The joint particle filtering and association is summarized in Algorithm 2. Similar ideas in the context of robot con-

Algorithm 2 SIR/MCJPDA estimation and association.

1: Set $t = 0$, generate N_t samples from each target $j = 1, \ldots, \tau$, i.e., $x_0 = \{x_0^{(i)}\}_{i=1}^{N_t} = \{x_0^{(i),1}, \ldots, x_0^{(i),\tau}\}_{i=1}^{N_t}$, where $x_0^{(i),j}$ from $p(x_0^j)$.
2: For each particle compute the weights for all measurement to track association $\gamma_t^{(i)} = \sum_{n=1}^{(M_t+1)^\tau} \text{Prob}(H_n^{(i)})$ and normalize for each measurement, i.e., $\tilde{\gamma}_t^{(i)} = \gamma_t^{(i)} / \sum_{i=1}^{N_t} \gamma_t^{(i)}$, where $\text{Prob}(H_n^{(i)})$ is the probability for hypothesis n using particle i according to equation (6).
3: Generate a new set $\{x_t^{(i\star)}\}_{i=1}^{N_t}$ by resampling with replacement N_t times from $\{x_t^{(i)}\}_{i=1}^{N_t}$, where $\text{Prob}(x_t^{(i\star)} = x_t^{(l)}) = \tilde{\gamma}_t^{(l)}$.
4: Predict (simulate) new particles, i.e., $x_{t+1}^{(i),j} = f(x_t^{(i\star),j}, w_t^{(i),j}), i = 1, \ldots, N_t$, using different noise realizations for the particles, for each target $j = 1, \ldots, \tau$.
5: Increase t and iterate to item 2.

trol appear in [19]. The optional particle number controller described in Section 3, is applied at step 3, in Table 2.

To simplify the algorithm some practical problems are discarded. The measurements within a scan is considered given at the same time instances and the number of targets (τ) is assumed constant during the simulation. If the number of targets is unknown or changing, the algorithm could be modified, for instance using a separate track start hypothesis. This could be done within the particle filter framework or possible to use some linearized method. To allow measurements with different time, the prediction step is modified with an increased computational load as a consequence, i.e., each track must be predicted to every measurement time, in the association step.

6 Simulations

In a simulation study, the proposed SIR/MCJPDA method is implemented for a multi target environment problem. The application at hand is a missile to air scenario. To simplify the simulations we assume that it is always possible to resolve

the targets. In Cartesian coordinates the relative state vector is defined as $x_t = \bar{x}_t - \bar{x}_t^{own}$, such that $x_t = \begin{pmatrix} X(t) & Y(t) & Z(t) & V_x(t) & V_y(t) & V_z(t) \end{pmatrix}^T$, where X, Y and Z are the Cartesian position coordinates and V_x, V_y and V_z the velocity components. The following discrete time system is used

$$x_{t+1} = \left(\begin{array}{c|c} \mathcal{I}_3 & T\mathcal{I}_3 \\ \hline \mathcal{O}_3 & \mathcal{I}_3 \end{array}\right) x_t + \begin{pmatrix} \frac{T^2}{2}\mathcal{I}_3 \\ \hline T\mathcal{I}_3 \end{pmatrix} w_t,$$

$$y_t = h(x_t) = \begin{pmatrix} \sqrt{X_t^2 + Y_t^2 + Z_t^2} \\ \arctan(\frac{Y_t}{X_t}) \\ \arctan(\frac{-Z_t}{\sqrt{X_t^2+Y_t^2}}) \end{pmatrix} + e_t,$$

where the process noise w_t is assumed Gaussian, $w_t \sim \mathcal{N}(0, Q)$. The three-by-three null matrix and unity matrix is denoted $\mathcal{O}_3$ and $\mathcal{I}_3$ respectively. The measurement noise is assumed Gaussian $e_t \sim \mathcal{N}(0, R)$. The parametric models for false alarms are assumed $N_{\text{FA}} \in Po(\lambda V)$, with average number of false alarms per unit volume λ and the validation region volume V. In the simulations $\mathbb{E}(N_{FA}) = \lambda V = 0.5$ is used. The detection probability is assumed $P_D = 0.9$. Assume the number of targets $\tau = 2$ and a sample time of $T = 1[s]$. The initial inertial target state vectors $\bar{x}_0^i$, initial own platform $\bar{x}_0^{own}$, measurement noise matrix R, process noise Q and initial state error matrix P_0 are

$$\bar{x}_0^1 = \begin{pmatrix} 6500 \\ -1000 \\ 2000 \\ -50 \\ 100 \\ 0 \end{pmatrix}, \bar{x}_0^2 = \begin{pmatrix} 5050 \\ -450 \\ 2000 \\ 100 \\ 50 \\ 0 \end{pmatrix}, \bar{x}_0^{own} = \begin{pmatrix} 0 \\ 0 \\ 3000 \\ 200 \\ -50 \\ 0 \end{pmatrix},$$

$$P_0 = \text{diag}\begin{pmatrix} 100^2 & 100^2 & 100^2 & 50^2 & 50^2 & 50^2 \end{pmatrix},$$

$$Q = \begin{pmatrix} 10^2 & 0 & 0 \\ 0 & 10^2 & 0 \\ 0 & 0 & 10^2 \end{pmatrix}, R = \begin{pmatrix} 50^2 & 0 & 0 \\ 0 & 0.01^2 & 0 \\ 0 & 0 & 0.01^2 \end{pmatrix}.$$

The implemented EKF is according to the *discretized linearization* technique [13], i.e., first linearize the underlying continuous time system and then discretize. Initial values for the tracks is draw from the initial uncertainty region P_0 around the true value. We compare the SIR/MCJPDA method with an NN data association where the estimation is done by the particle filter and where the covariance matrix needed for the association is similar to equation (5). A comparison is also made to an EKF using the NN or JPDA association in a similar way. In Figure 2, a data association and estimation using the SIR/MCJPDA filter is presented. To evaluate the performance a *root mean square error* (RMSE) analysis is performed over $N_{mc} = 60$ simulations and time samples. In Table 1, the results for the different methods are summarized, using RMSE for the two targets when $t \geq 3$, ignoring initial transients. The particle filter used $N = 25000$ samples. In Figure 3, the RMSE

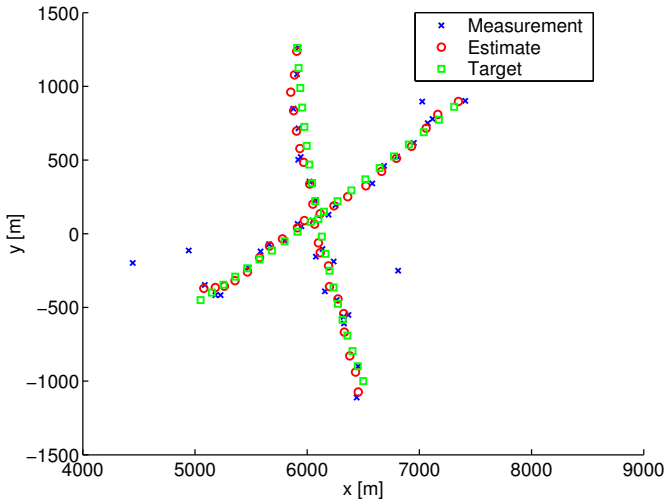


Figure 2: *Data association & tracking.*

Table 1: *Association & estimation – RMSE analysis.*

Estimation	**Association**	**RMSE #1**	**RMSE #2**
SIR	MCJPDA	51.7	51.4
SIR	NN	55.9	55.5
EKF	JPDA	52.1	51.5
EKF	NN	52.7	54.0

values for different times are presented for the methods described in Table 1 (target 1). In Figure 4, the particle number controller (Section 3), for SIR/MCJPDA is used with $k_1 = \frac{1}{2}, k_2 = 0.1, \Lambda = 9.5$ and $\alpha_{\text{inc}}(N_t) = 0.2N_t, \alpha_{\text{dec}}(N_t) = -0.1N_t$, for the marginal case, for 20 Monte Carlo simulations.

7 Conclusions

In this paper a novel Monte Carlo data association method for jointly estimation and association in a probabilistic data association framework is presented. This method (SIR/MCJPDA) is compared to EKF based classical association methods such as NN and JPDA. The NN association is also applied to the SIR method, where the covariance is calculated from the particle filter cloud. A novel approach to determine the number of particles for each target is also developed, using a relay and an integrator in a feedback system. In the simulation study in Section 6, the methods are compared and the RMSE is used to describe the performance.

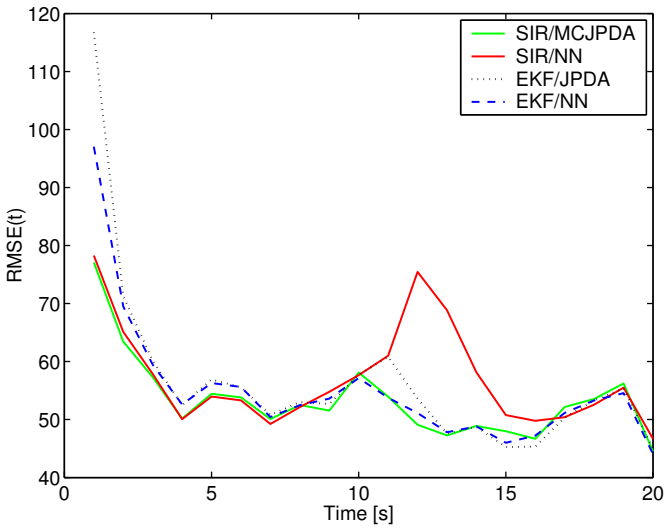


Figure 3: *RMSE(t) for different methods.*

For more nonlinear problems and problems where the noise distribution is highly non-Gaussian, the proposed simulation based algorithms may increase the overall tracking performance.

References

[1] B.D.O. Anderson and J. B. Moore. *Optimal Filtering.* Prentice Hall, Englewood Cliffs, NJ, 1979.

[2] D. Avitzour. Stochastic simulation Bayesian approach to multitarget tracking. *IEE Proceedings on Radar, Sonar and Navigation*, 142(2), 1995.

[3] Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*, volume 179 of *Mathematics in Science and Engineering.* Academic Press, 1988.

[4] Y. Bar-Shalom and X. R. Li. *Estimation and Tracking: Principles, Techniques, and Software.* Artech House, 1993.

[5] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications.* Linköping Studies in Science and Technology. Dissertations No. 579, Linköping University, Linköping, Sweden, 1999.

[6] N. Bergman and A. Doucet. Markov Chain Monte Carlo Data Association for Target Tracking. In *Proceedings IEEE Conference on Acoustics, Speech and Signal Processing*, 2000.

[7] S. S. Blackman. *Multiple-Target Tracking with Radar Applications.* Artech House, Norwood, MA, 1986.

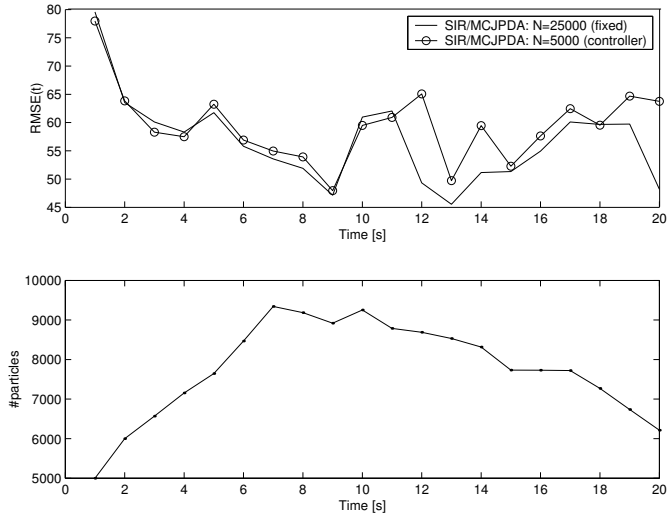


Figure 4: *The particle number controller.*

[8] S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[9] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[10] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[11] N. J. Gordon. A hybrid bootstrap filter for target tracking in clutter. In *IEEE Transactions on Aerospace and Electronic Systems*, volume 33, pages 353–358, 1997.

[12] N. J. Gordon, D. J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[13] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons Ltd, 2000.

[14] C. Hue, J. P. Le Cadre, and P. Pérez. Tracking multiple objects with particle filtering. Technical Report Research report IRISA, No. 1361, October 2000.

[15] C. Rago, P. Willett, and R. Streit. A comparison of the JPDAF and PMHT tracking algorithms. In *Proceedings IEEE Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 3571–3574, 1995.

[16] D. B. Reid. The application of multiple target tracking theory to ocean surveillance. In *Proceedings of the 18:th IEEE Conference on Decision and Control*, pages 1046–1052, Fort Lauderdale, FL, USA, 1979.

[17] B. D. Ripley. *Stochastic Simulation*. John Wiley, 1988.

[18] D. J. Salmond, D. Fisher, and N. J. Gordon. Tracking in the presence of intermittent spurious objects and clutter. In *SPIE Conference on Signal and Data Processing of Small Tragets*, 1998.

[19] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *IEEE Proceedings International Conference on Robotics and Automation*, volume 2, pages 1665–1670, 2001.

Paper G

Auxiliary Particle Filters for Tracking a Maneuvering Target

Edited version of the paper:

R. Karlsson and N. Bergman. Auxiliary particle filters for tracking a maneuvering target. In *Proceedings of the 39:th IEEE Conference on Decision and Control*, pages 3891–3895, Sydney, Australia, December 2000.

Auxiliary Particle Filters for Tracking a Maneuvering Target

Rickard Karlsson and Niclas Bergman

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

Abstract

We consider the recursive state estimation of a highly maneuverable target. In contrast to standard target tracking literature we do not rely on linearized motion models and measurement relations, or on any Gaussian assumptions. Instead, we apply optimal recursive Bayesian filters directly to the nonlinear target model. We present novel sequential simulation based algorithms developed explicitly for the maneuvering target tracking problem. These Monte Carlo filters perform optimal inference by simulating a large number of tracks, or particles. Each particle is assigned a probability weight determined by its likelihood. The main advantage of our approach is that linearizations and Gaussian assumptions need not be considered. Instead, a nonlinear model is directly used during the prediction and likelihood update. Detailed nonlinear dynamics models and non-Gaussian sensors can therefore be utilized in an optimal manner resulting in high performance gains. In a simulation comparison with current state-of-the-art tracking algorithms we show that our approach yields performance improvements. Moreover, incorporation of physical constraints with sustained optimal performance is straightforward, which is virtually impossible to incorporate for linear Gaussian filters. With the particle filtering approach we advocate these constraints are easily introduced and improve the results.

Keywords: Maneuvering target tracking, Auxiliary particle filter, IMM.

1 Introduction

Traditionally, target tracking problems are solved using linearized tracking filters, mainly *extended Kalman filters* (EKFs) [1]. For highly maneuvering targets or for low observation rates different maneuvering modes are used to describe the motion. Therefore, target maneuvers are often described by multiple linearized models. A

common method is the *Interacting Multiple Model* filter (IMM) [6]. State-of-the-art estimation and tracking literature [2], present state estimation and prediction performed by switching between models or by mixing them. If the filter update is slow or the target maneuver large the linearized solution may not always be good. Nonlinear models in state equation and measurement relation and a non-Gaussian noise assumption may also lead to non optimal solutions. Moreover, the incorporation of constraints to the system parameters is complicated using linearized techniques.

The sequential Monte Carlo methods, or particle filters, provide general solutions to many problems where linearizations and Gaussian approximations are intractable or would yield too low performance. The maneuvering target tracking problem is an application which has strong elements of nonlinearity. Non-Gaussian noise assumptions and incorporation of constraints on some of the system parameters can also be performed in a natural way by using these simulation based methods. The constraints are due to limitations in state variables but could also be induced by the terrain, such as land avoidance for tracking ships. Monte Carlo techniques have been a growing research area lately due to improved computer performance. Particularly some aspects of the bearings-only tracking application has been investigated [3].

In this paper we extend the *auxiliary particle filter* (APF) of Pitt and Shepard [8] to the case of multiple nonlinear models, switching according to a Markov transition kernel. Each particle is split deterministically into a number of possible maneuver hypotheses and the likelihood is adjusted by the Markov transition probabilities.

The algorithm is implemented for the classical *Bayesian Bootstrap* method [5] using the Auxiliary Particle method. In a simulation study we compare this filter to a linearized method using an *interacting multiple model* filter (IMM) [6] based on three *extended Kalman filters* (EKFs) for an *air traffic control* (ATC) *track-while-scan* (TWS) application. The problem under consideration incorporates nonlinear effects both in the prediction and measurement model and some constraints on the system states.

2 The Target Tracking Model

A general target tracking problem consists of a nonlinear state equation and a nonlinear measurement relation of the form

$$\begin{aligned} x_{t+1} &= f(x_t, w_t), \\ y_t &= h(x_t, e_t), \end{aligned}$$

where the process noise w_t and measurement noise e_t are non-Gaussian, describing the target maneuver and measurement. Some states or combination thereof are in general constrained by some parameter dependent set $\Lambda_i(x)$. The constraints of states are due to target maneuvering capabilities or to terrain constraints. In this paper only constraint of the target speed is considered among a state dependent

turn rate. The constraint is defined in discrete time, but a continuous constraint could also be possible, if a numerical solution to the state equation is used.

Following standard target tracking literature [2], tracking an aircraft in nearly coordinated flight using a radar sensor yields a model with nonlinear elements both in the dynamic state equation and in the measurement relation. The model is a discretized continuous time nonlinear stochastic differential equation model where the turn rate state ω gives a strong nonlinear behavior. For highly maneuvering targets we extended it with a maneuvering model where the target maneuver is described by a Markovian switching structure. The amount of the turn rate is assumed velocity dependent due to the fact that the pilot will induce a moderate turn rate traveling at high speed. The turn rate is modeled so a change will be visible in the measurement directly following the onset of a turn, and modeled as a set of three discrete values for right turn, straight flying or left turn. The discrete system is given by

$$\chi_{t+1} = A(\omega_{t+1})\chi_t + \begin{pmatrix} B_w & B_\omega \end{pmatrix} w_t, \tag{1}$$

$$\chi_t = \begin{pmatrix} x_t & \omega_{t+1} \end{pmatrix}^T, x_t = \begin{pmatrix} X & \dot{X} & Y & \dot{Y} \end{pmatrix}^T, \tag{2}$$

where X and Y is the Cartesian position coordinates and $\dot{X}, \dot{Y}$ the velocity components. The velocity is constrained to some set V, i.e., $\sqrt{\dot{X}^2 + \dot{Y}^2} \in V$.

$$A(\omega) = \begin{pmatrix} 1 & \frac{\sin(\omega T)}{\omega} & 0 & -\frac{(1-\cos(\omega T))}{\omega} & 0 \\ 0 & \cos(\omega T) & 0 & -\sin(\omega T) & 0 \\ 0 & \frac{(1-\cos(\omega T))}{\omega} & 1 & \frac{\sin(\omega T)}{\omega} & 0 \\ 0 & \sin(\omega T) & 0 & \cos(\omega T) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$B_v = \begin{pmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \\ 0 & 0 \end{pmatrix}, B_\omega = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

The turn rate is assume velocity dependent according to the following model $\omega = a_{typ}/\sqrt{\dot{X}^2 + \dot{Y}^2}$, where a_{typ} is the typical maneuvering acceleration which is modeled as a set of three discrete values, having a Markovian switching structure. The radar measurements are modeled as

$$y_t = h(x_t) + e_t = \begin{pmatrix} \sqrt{X^2 + Y^2} \\ \arctan(\frac{Y}{X}) \end{pmatrix} + e_t,$$

where e_t is zero mean noise with covariance R_t. Independence in time and between the measurement and process noise is assumed.

In the Bayesian bootstrap algorithm presented in section 4 the prediction stage is performed by simply simulating equation (1) with the noise model. For systems with an explicit solution this yields a simple and efficient predictor.

3 Particle Filters

Many engineering problems are by nature recursive and require on-line solutions. Common applications such as state estimation, recursive identification and adaptive filtering often require recursive solutions to problems having both nonlinear and non-Gaussian character. The seminal paper of Gordon et al. [5] marks the onset of a rebirth for algorithms based on Monte Carlo simulation techniques for solving this important class of problems in an optimal manner.

The sequential Monte Carlo methods, or particle filters provide an approximative Bayesian solution to discrete time recursive identification or filtering problems by updating an approximative description of the posterior filtering density. Let x_t denote the state of the observed system and $\mathbb{Y}_t = \{y_i\}_{i=1}^t$ be the set of observed measurements until present time. The Monte Carlo filter approximate the density $p(x_t|\mathbb{Y}_t)$ by a large set of N particles $\{x_t^{(i)}\}_{i=1}^N$, where each particle has an assigned relative weight, $\gamma_t^{(i)}$, chosen such that all weights sum to unity. The location and weight of each particle reflect the value of the density in the region of the state space, $p(x_t|\mathbb{Y}_t) \approx \frac{1}{N}\sum_{i=1}^N \delta(x_t - x_t^{(i)})$. The particle filter updates the particle location and the corresponding weights recursively with each new observed measurement.

The nonlinear prediction density $p(x_t|\mathbb{Y}_{t-1})$ and filtering density $p(x_t|\mathbb{Y}_t)$ for the Bayesian inference is given by

$$p(x_t|\mathbb{Y}_{t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|\mathbb{Y}_{t-1})dx_{t-1}, \tag{3}$$

$$p(x_t|\mathbb{Y}_t) \propto p(y_t|x_t)p(x_t|\mathbb{Y}_{t-1}). \tag{4}$$

The main idea is to approximate $p(x_t|\mathbb{Y}_{t-1})$ with

$$p(x_t|\mathbb{Y}_{t-1}) \approx \frac{1}{N}\sum_{i=1}^N \delta(x_t - x_t^{(i)}).$$

Inserting into (4) yields a density to sample from. This can be done by using the Bayesian bootstrap or *sampling importance resampling* (SIR) algorithm from [5].

Algorithm 1 Sampling Importance Resampling (SIR)

1: Set $t = 0$, and generate N samples $\{x_0^{(i)}\}_{i=1}^N$ from $p(x_0)$.
2: Compute the likelihood weights $\gamma_t^{(i)} = p(y_t|x_t^{(i)})$ and normalize, i.e., $\tilde{\gamma}_t^{(i)} = \gamma_t^{(i)}/\sum_{j=1}^N \gamma_t^{(j)}, i = 1, \ldots, N$.
3: Generate a new set $\{x_t^{i\star}\}_{i=1}^N$ by resampling with replacement N times from $\{x_t^{(i)}\}_{i=1}^N$, where $\text{Prob}(x_t^{(i\star)} = x_t^{(j)}) = \tilde{\gamma}_t^{(j)}$.
4: Predict (simulate) new particles, i.e., $x_{t+1}^{(i)} = f(x_t^{(i\star)}, w_t^{(i)}), i = 1, \ldots, N$ using different noise realizations for the particles.
5: Increase t and iterate to item 2.

4 Auxiliary Particle Filter for Maneuvering Targets

In this paper we are interested in what performance one can achieve by using true nonlinear inference based on Monte Carlo algorithms, applied to a target tracking application. In contrast to what is standard procedure in target tracking we choose not to linearize the target models for each mode. Instead, we apply particle filters directly to the mode switching model. The maneuvers are modeled by a discrete parameter ω (turn rate) with a Markovian switching structure yielding a mode switching, or jumping, model. In [7] a similar idea is used for linear jump Markov models, applied to a bootstrap filter. By using particle filters, nonlinearities in the system and measurement models and constraints in system parameters are incorporated in a natural way.

For reliable handling of the mode hypotheses we extend the state-of-the-art particle filter of Pitt and Shepard [8]. These so called auxiliary particle filters use resampling on the predicted particles to select which particles to use in the prediction and measurement update. We introduce a deterministic splitting of each particle into several offspring, each one representing a different target maneuver. Each offspring is weighted by the Markov transition probability for the maneuver and its likelihood. In Figure 1 an example of three different maneuver assumptions in the deterministic particle splitting is visualized in the upper plot. The three predicted particle clouds conditioned on the turn rate are clearly distinguished in the figure. The resampled cloud using the predicted particles is viewed in the lower plot. In the lower plot in Figure 1 the resampled particles are predicted one step forward.

System constraints are also incorporated in the model, so that non feasible maneuvers are avoided using the particle filtering technique.

The Auxiliary Particle Filter [8] extends the state x_t by predicting the state conditional upon particle k. At time t, the particle set $\{x_t^{(k)}\}_{k=1}^N$ and the corresponding weights $\pi_t^{(k)}$ form the following approximations to the prediction and filter densities for the state of the target

$$\begin{aligned} p(x_{t+1}|\mathbb{Y}_t) &= \sum_{j=1}^{N} p(x_{t+1}|x_t^{(j)})\pi_t^{(j)}, \\ p(x_{t+1}|\mathbb{Y}_{t+1}) &\propto p(y_{t+1}|x_{t+1}) \sum_{j=1}^{N} p(x_{t+1}|x_t^{(j)})\pi_t^{(j)}. \end{aligned}$$

By defining

$$p(x_{t+1}, k|\mathbb{Y}_{t+1}) \propto p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t^{(k)})\pi_t^{(k)}, k = 1, \ldots, N,$$

we can draw from this joint density and then discard the index, to produce a sample from the empirical filtering density as required. The index k is referred to as an auxiliary variable. For the target tracking application we consider the joint density

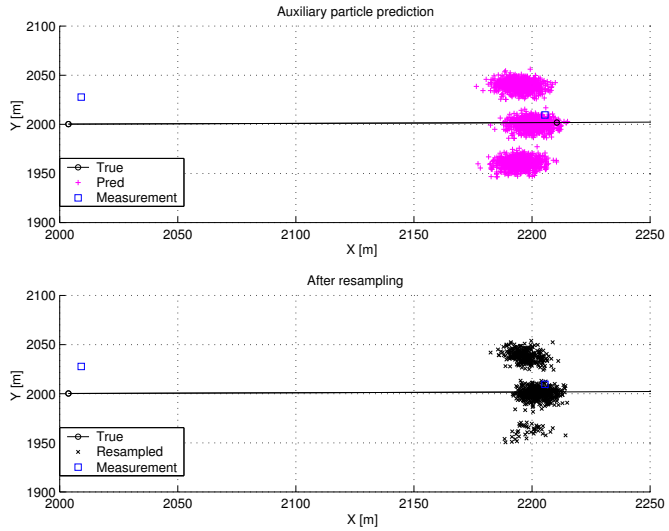


Figure 1: *The three different maneuvers, Auxiliary particle filter.*

of particle k at time t and the target state at time $t+1$. The turn rate ω_{t+1} indicates that the state is effected during the integration of the state equation from t to $t+1$, i.e., it will effect the states at time $t+1$, when applied at time t. Using Baye's rule gives

$$\begin{aligned} p(x_{t+1}, \omega_{t+1}, k|\mathbb{Y}_{t+1}) &\propto p(y_{t+1}|x_{t+1}, \omega_{t+1}, k)p(x_{t+1}, \omega_{t+1}, k|\mathbb{Y}_t) \\ &= p(y_{t+1}|x_{t+1})p(x_{t+1}|\omega_{t+1}, k, \mathbb{Y}_t)p(\omega_{t+1}, k|\mathbb{Y}_t) \\ &= p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t^{(k)}, \omega_{t+1})p(\omega_{t+1}|k, \mathbb{Y}_t)p(k|\mathbb{Y}_t) \\ &= p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t^{(k)}, \omega_{t+1})p(\omega_{t+1}|\omega_t^{(k)})\pi_t^{(k)}. \end{aligned} \tag{5}$$

Approximating this expression by replacing x_{t+1} with the expected mean

$$\mu_{t+1}^{(k)}(\omega_{t+1}) = \mathbb{E}\left(x_{t+1}|\omega_{t+1}, x_t^{(k)}\right),$$

in the first factor gives

$$p(x_{t+1}, \omega_{t+1}, k|\mathbb{Y}_{t+1}) \approx Cp(y_{t+1}|\mu_{t+1}^{(k)}(\omega_{t+1}))p(x_{t+1}|x_t^{(k)}, \omega_{t+1})p(\omega_{t+1}|\omega_t^{(k)})\pi_t^{(k)}.$$

Marginalization over x_{t+1} yields

$$p(\omega_{t+1}, k|\mathbb{Y}_{t+1}) \approx Cp(y_{t+1}|\mu_{t+1}^{(k)}(\omega_{t+1}))p(\omega_{t+1}|\omega_t^{(k)})\pi_t^{(k)}, \tag{6}$$

where C is a normalization factor. Sampling from the density (5) can now be performed by resampling with replacement from the set $\{x_t^{(k)}\}_{k=1}^N$, where the index

Algorithm 2 Auxiliary Particle Filter for maneuvering target tracking (APF)

1: Set $t = 0$, and generate N samples $\{x_0^{(i)}\}_{i=1}^N$ from $p(x_0)$, set $\mu_t^{(k)} = x_t^{(k)}$, $k^{(j)} = j, j = 1, \ldots, N$.
2: Compute $\mu_{t+1}^{(k)}(\omega_{t+1}) = \mathbb{E}\left(x_{t+1}|x_t^{(k)}, \omega_{t+1}\right)$ for every $\omega_{t+1} \in \mathcal{M}(x_t^{(k)})$, where $\mathcal{M}(x_t^{(k)})$ is the set of all feasible (state dependent) maneuvers. Number of particles after splitting: M.
3: Generate new indices $k^{(j)}$ by sampling N times from $p(k|\mathbb{Y}_{t+1}) \propto \pi_t^{(k)} p(y_{t+1}|\mu_{t+1}^{(k)}) p(\omega_{t+1}|\omega_t^{(k)})$ and predict (simulate) the particles, i.e., $x_{t+1}^{(j)} = f(x_t^{(k^{(j)})}, w_t^{(j)}), j = 1, \ldots, N$ with different noise realizations.
4: Compute the likelihood weights $\gamma_t^{(j)} = \frac{p(y_{t+1}|x_{t+1}^{(j)})}{p(y_{t+1}|\mu_{t+1}^{(k^{(j)})})}$ for $j = 1, \ldots, N$ and normalize, i.e., $\pi_t^{(j)} = \frac{\gamma_t^{(j)}}{\sum_{j=1}^M \gamma_t^{(j)}}$
5: Perform an optional resampling of the set $\{x_{t+1}^{(i)}\}_{i=1}^N$, using the probability weights. If resampling is chosen then reset $\pi_t^{(j)} = \frac{1}{N}, j = 1, \ldots, N$.
6: Increase t and iterate to item 2.

is chosen proportional to (6). The resampled candidates are then predicted using the system model. In summary, the algorithm is given by

Implementing three different maneuvers for left turn, straight flying or right turn gives $M = 3N$ in the algorithm, which will only marginally increase the computational burden, since the Bayesian bootstrap method with/without auxiliary particle filters is linear in complexity $\mathcal{O}(M)$. This is because the resampling stage can be done in a fast way by using a classical algorithm for sampling M ordered independent identically distributed variables [4],[9].

5 Simulations

A simulation study using the nearly coordinated turn model from section 2 is performed. The maneuvering is done by setting up a flight path in accordance with the used Markov transition density. Simulations are performed by using the APF extended with the deterministic maneuver splitting discussed in section 4. A comparison to a traditional tracking method based on an IMM-filter consisting of three EKFs with different turn assumptions is made. In the simulation study the only parameter constraint considered is a limitation on the target speed, to the interval $50 \leq |v| \leq 60$ m/s. The simulation step is re-run until a feasible speed is achieved. The distribution of the measurement noise is chosen to be Gaussian, with angular and distance standard deviations of 0.5^o respectively 20 m. The sampling period is chosen to $T = 4$ s to emulate a *track-while-scan* (TWS) behavior. For modern mono-pulse radars the update time and the angle standard deviation may be much smaller, but for non-monopulse system or for air-borne tracking systems

much greater values should be used.

In Figure 2 one realization using Gaussian noise is viewed. The a posteriori probabilities for each coordinate is presented in Figure 3 for the predicted particles for one realization.

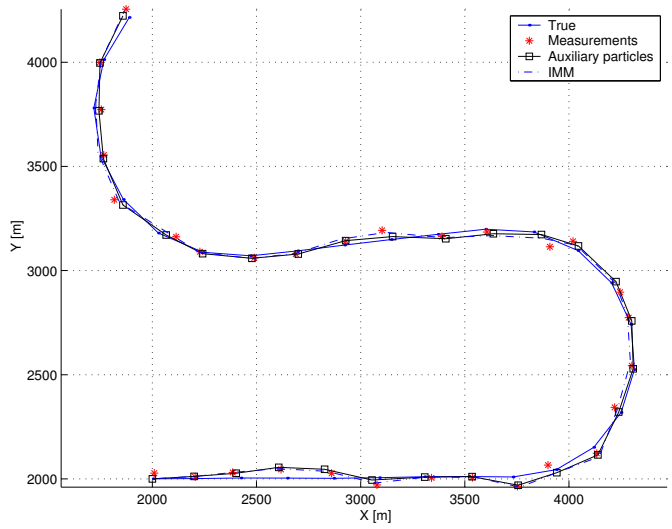


Figure 2: *Simulation overview.*

In Table 1 the position *root mean square error* (RMSE) for the APF with $N = 800$ particles is compared to the IMM-filter, using $N_{\text{MC}} = 100$ Monte Carlo simulations. The RMSE using measurements only is also presented in Table 1. The RMSE is defined by equation (7). As seen the APF method improves tracking performance.

$$\text{RMSE} = \sqrt{\frac{1}{L}\sum_{t=1}^{L}\frac{1}{N_{\text{MC}}}\sum_{i=1}^{N_{\text{MC}}}(\hat{X}_t^{(i)} - X_t^{\text{TRUE}})^2 + (\hat{Y}_t^{(i)} - Y_t^{\text{TRUE}})^2}, \tag{7}$$

where $L = 30$ is the simulation path length and $\hat{X}_t^{(i)}, \hat{Y}_t^{(i)}$, are the filter position estimates at time t in Monte Carlo run i.

Table 1: *RMSE for 100 Monte Carlo simulations, using $N = 800$ particles.*

	APF	**IMM**	**Measurements**
RMSE	32.09	37.05	41.5

In Figure 4 the RMSE is presented for each time, i.e., according to (8) for the

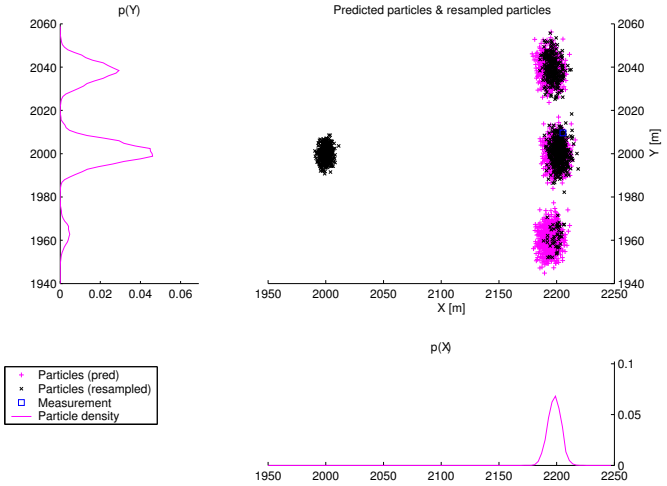


Figure 3: *Auxiliary particles & posterior PDF.*

different methods

$$\mathrm{RMSE(t)} = \sqrt{\frac{1}{N_{\mathrm{MC}}} \sum_{i=1}^{N_{\mathrm{MC}}} (\hat{X}_t^{(i)} - X_t^{\mathrm{TRUE}})^2 + (\hat{Y}_t^{(i)} - Y_t^{\mathrm{TRUE}})^2}. \tag{8}$$

6 Conclusions

In the simulation study in section 5 the auxiliary particle filter method improved tracking performance compared to the IMM-filter for the track-while-scan air traffic control application. The particle filter method is also more flexible than traditional methods since it can also incorporate system constraints and non-Gaussian noise assumptions. However, simulation based methods such as particle filters can be time consuming if many particles are used. To improve the real-time execution performance the particle filter update could be run in parallel.

References

[1] B.D.O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, NJ, 1979.

[2] Y. Bar-Shalom and X. R. Li. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, 1993.

[3] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for non-linear problems. In *IEE Proceedings on Radar and Sonar Navigation*, 1999.

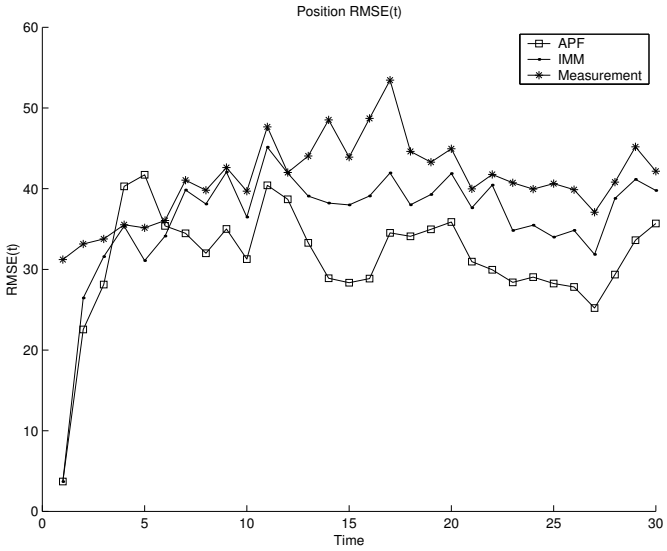


Figure 4: *RMSE(t) for different tracking methods.*

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

[5] N. J. Gordon, D. J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[6] X. R. Li and Y. Bar-Shalom. Design of an interactive multiple model algorithm for air traffic control tracking. *IEEE Transactions on Control Systems Technology*, 1:186–194, 1993.

[7] S. McGinnity and G. Irwin. Multiple model estimation using the bootstrap filter. In *IEE Colloquium on Target Tracking and Data Fusion (Digest No. 1998/282)*, 1998.

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

[9] B. D. Ripley. *Stochastic Simulation*. John Wiley, 1988.

Paper H

Model-Based Statistical Tracking and Decision Making for Collision Avoidance Application

Edited version of the paper:

R. Karlsson, J. Jansson, and F. Gustafsson. Model-based statistical tracking and decision making for collision avoidance application. In *Proceedings of American Control Conference*, Boston, MA, USA, June 2004.

Model-based Statistical Tracking and Decision Making for Collision Avoidance Application

Rickard Karlsson, Jonas Jansson, and Fredrik Gustafsson,

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

Abstract

A growing research topic within the automotive industry is active safety systems. These systems aim at helping the driver avoid or mitigate the consequences of an accident. In this paper a collision mitigation system that performs late braking is discussed. The brake decision is based on estimates from tracking sensors. We use a Bayesian approach, implementing an extended Kalman filter (EKF) and a particle filter to solve the tracking problem. The two filters are compared for different sensor noise distributions in a Monte Carlo simulation study. In particular a bi-modal Gaussian distribution is proposed to model measurement noise for normal driving. For ideal test conditions the noise probability density is derived from experimental data. The brake decision is based on a statistical hypothesis test, where collision risk is measured in terms of required acceleration to avoid collision. The particle filter method handles this test easily. Since the test is not analytically solvable a stochastic integration is performed for the EKF method. Both systems perform well in the simulation study under the assumed sensor accuracy. The particle filter based algorithm is also implemented in a real-time testbed and fullfilled the on-line requirements.

Keywords: Collision mitigation by braking, Particle filter, EKF, Hypothesis test.

1 Introduction

A current trend in automotive industry is to introduce active safety systems that avoid or mitigate collisions. One system with a potential large positive impact

on accident statistics is *forward collision avoidance systems* (FCAS), using sensors such as radar, lidar and cameras to monitor the region in front of the vehicle. A tracking algorithm is used to estimate the state of the objects ahead and a decision algorithm uses the estimated states to determine any action. We will specifically look at a system that performs late braking to reduce the collision speed, which is referred to as *collision mitigation by braking system* (CMbBS). This type of system is discussed further in [6]. There are several motivations for this kind of system. One is its potential ability to affect rear-end collisions which constitute approximately 30%, see [13], of all collisions. Furthermore human factors contribute to approximately 90% of all traffic accidents, [12]. For reasons such as driver acceptance and legal requirements of the system, the tracking and decision algorithms are crucial. Current state of the art automotive tracking algorithms use Kalman or extended Kalman filter (EKF) due to computational cost.

In this paper we will propose a new method for collision avoidance applications based on Bayesian estimation methods and hypothesis testing. Two tracking algortihms are tested. One based on EKF an one on the particle filter. The latter method is also implemented in a real-time test system and its real-time feasability is demonstrated in spite of its computer intensive complexity compared to the EKF.

2 Tracking and Vehicle Models

2.1 Tracking Model

The general tracking model takes the form of a nonlinear state space model for the vehicle dynamics and sensor measurements

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

$$y_t = h(x_t) + e_t. \tag{1b}$$

For our radar based CMbB system, the state vector, $x_t \in \mathbb{R}^n$, contains relative position to the other vehicle of potential collision risk, and relative velocity. Here both longitudinal and lateral directions are used. Further, u_t is the known inputs from the accelerator, brake and steering wheel. These are also present for the target vehicle, but then un-measureable and hence treated as state noise, w_t. The measurement relation (1b) comes from radar measurements of range, range rate and bearing. The measurement noise, e_t, includes clutter, multi-path and multiple reflection points in the vehicle ahead.

2.2 Estimation Approaches

For the CMbB system the measurement noise and the process noise (driver inputs) are not necessarily Gaussian. Thus we need a recursive estimation method that can handle this. In this paper we will study a general nonlinear and non-Gaussian

Bayesian estimation problem. This is in general non-tractable, but using the particle filter method [4] (see Section 4.1) a recursive solution to the problem is given. We will also compare with the linearized solution using the extended Kalman filter (EKF).

2.3 State Noise Model

The classical assumption in target tracking is to assume a Gaussian distribution for w_t. In Figure 1 recorded driving data provided by Volvo is used to make histograms of the longitudinal component of the acceleration noise, w_t. The data is of course highly correlated in time, since the acceleration does not fluctate rapidly. It could be used by introducing a Markov chain based on the empirical data. However, for braking situations probably a distribution based on accelerations collected close to rapid decelerations should be used instead. A further and natural option is

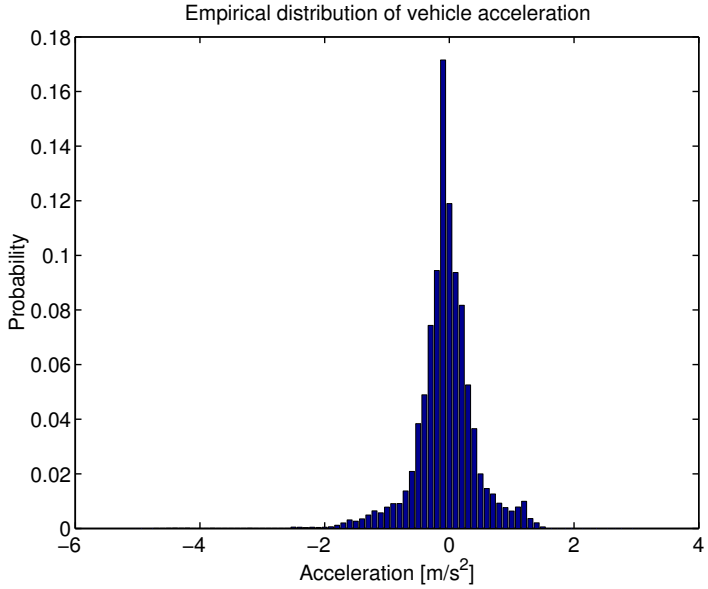


Figure 1: *Acceleration histogram of a vehicle driven in urban traffic for 45 minutes.*

to switch between different distributions depending on the driving situation. In urban traffic one distribution is used, on highway another one, *etc.* This leads to a mode dependent distribution $p_w(w;\xi)$. Here the mode can be determined from the host vehicle speed, it may depend on the state vector, external information from navigation and telematics systems *etc.* A simple approach is to use one mode when $v < 70$ km/h and another one for $v > 70$ km/h.

2.4 Measurement Noise Model

The measurement equation for the radar involves range, range rate and relative angle measurements to the vehicle ahead. Similarly to the above, the range error distribution can be modeled. A common model is to assume Gaussian noise. However, we will extend this to a more general case. An experiment is performed by towing a radar equipped vehicle, which can then measure the range to the towing vehicle under driving conditions with an almost constant range. Data is collected using an FM-CW radar at 40 m and constant speed 60 km/h on a Volvo S80 sedan. The range data is correlated, probably due to some oscillations in the rope and from internal filters in the receiver. A second order state space model is fitted to data using the `n4sid` method, [11, 10], in order to produce de-correlated range data. This pdf range error is presented in Figure 2 together with a Gaussian approximation.

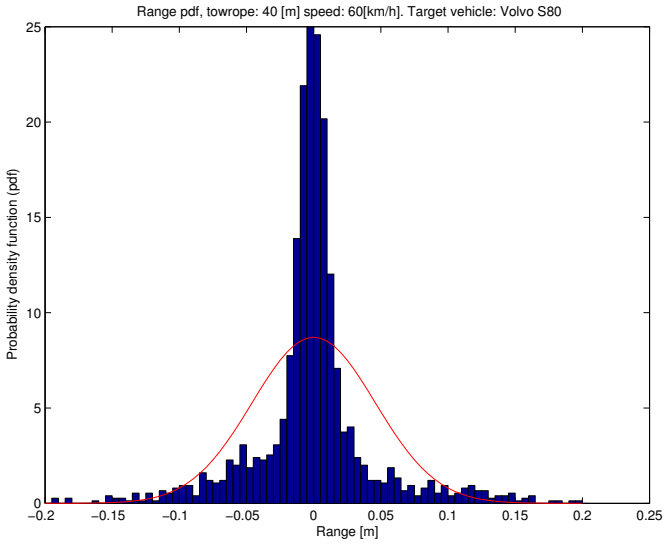


Figure 2: *Measurement range noise pdf around the nominal range $r = 40$ m for the target vehicle Volvo S80 sedan and an equivalent Gaussian approximation.*

The measurement noise from the experiment is collected under nearly ideal circumstances. For instance, the data was collected from behind, i.e., no aspect angle to the car, and on a smooth test track. In many cases due to road curvatures, uneven road surfaces and lateral movement the measurement range distribution might be different. Also, the exact reflection point is uncertain since the azimuth angle is not that exact, so for medium distances, the main reflection point may be located on different parts of the car. Hence, there are indications that the measure-

ment noise pdf has a larger variance and different shape. A bi-modal Gaussian pdf example is presented in Figure 3. This measurement noise model is motivated by modeling the vehicle by two point reflectors, which is a simplified way of describing both multi-path propagation and complex reflection geometry. The exact appearance of the pdf will vary with many factors such as target vehicle, sensor and traffic situation. Here we only test a bi-modal distribution with highly separated peaks. This is to cover many situations with different aspect angles, where the density is due to multi-path propagation, and multiple reflections points for instance in the rear-end, wheel housing and rear-view mirror. More measurements and experiments are needed to establish the empirical pdf for different driving situations, so the proposed density should just be an example of non-Gaussian pdf influence. In the simulation study we will compare different assumptions of the pdf.

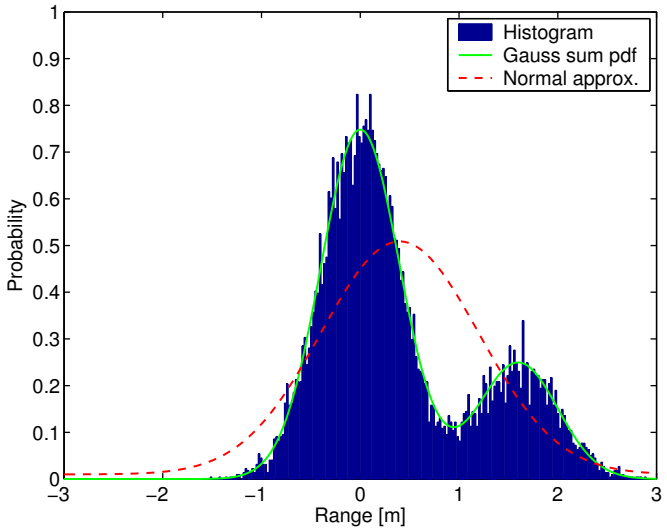


Figure 3: *Bimodal range measurement noise and Gaussian approximation of range density* $p_e = 0.75\mathcal{N}(0, 0.4^2) + 0.25\mathcal{N}(1.6, 0.4^2)$.

3 Statistical Decision Making

Statistical decision making requires an estimate of the distribution of x_t from noisy measurements y_t. The idea is for instance to calculate the probability for impact for each time. More on this for automotive application can be found in [6].

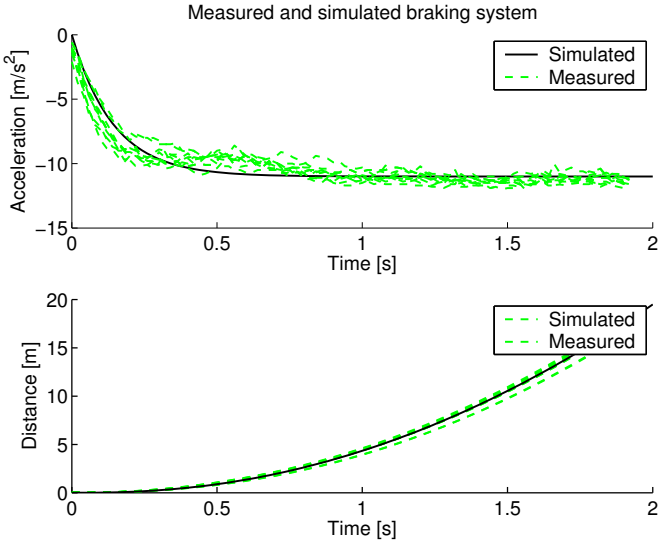


Figure 4: *Data from a hard braking maneuver, 110-0 km/h for a Volvo V70.*

3.1 Hypothesis Testing

A general model-based statistical decision rule based on some criterion, $g(x_t)$, takes the form

$$\text{Prob}(g(x_t) > 0) > 1 - \alpha, \tag{2}$$

where α determines the confidence level. Hence, in (2) one has to compute some integrals over the state-dimension to calculate the probability, given that we know the probability density function or an estimate thereof. Note that this is a more flexible approach than a simple rule like $g(\hat{x}_t) > 0$, since in (2) we use the complete *a posteriori* distribution of the state vector and not only the estimate $\hat{x}_t$.

How successful the rule (2) is in mitigating a collision depends on the chosen rule, but also on how accurately the estimated pdf reflects the true distribution. Almost always an analytical expression of the probability is hard or impossible to compute. One way is to use a numerical integration or more preferrable a stochastic integration to compute an approximation. Here, one takes a large number of samples $x_t^{(i)}$, $i = 1, 2, \ldots, N$ from the distribution $p(x_t|y_0, ..., y_t)$, and then

$$\text{Prob}(g(x_t) > 0) \approx \frac{\#g(x_t^{(i)}) > 0}{N}. \tag{3}$$

For the EKF a stochastic integration can be applied by drawing samples from the Gaussian distribution around the estimate $\hat{x}_t$. This procedure is much simpler when

the particle filter is used, since the samples, $x_t^{(i)}$, exist internally in the algorithm. One can simply count the number of particles for which the inequality is satisfied. This fact decreases the gap in computational burden between the EKF and particle filter approaches.

3.2 The CMbB Decision Rule

In this paper we will study a CMbB system, which performs maximum braking when a collision is becoming imminent, i.e., when a brake maneuver close to or exceeding the vehicles handling limits is needed to reach a zero speed at impact. We will introduce a very simple brake criteria to test the decision idea. We do this by calculating the required acceleration to obtain a zero velocity at a possible impact. Suppose at time $t = t_0$ the relative position and speed to the object ahead is given by p_0 and v_0. Denote the host vehicle acceleration by $u_t = a_{host}$ (input signal) and the target vehicle acceleration by a_{obj}. We consider both of these accelerations to be constant from time $t > t_0$ and denote the relative acceleration by $a = a_{host} - a_{obj}$. In the following analysis an ideal brake system is used for simplicity. For a high relative velocity the error of neglecting brake system characteristics can be compensated by adjusting the threshold in (7). The collision speed, v^c at the collision time t^c, can be calculated as $v^c = v_0 + a \cdot (t^c - t_0)$. The relative time to zero collision speed, $v^c = 0$, is hence

$$t^z = t^c - t_0 = -\frac{v_0}{a}. \tag{4}$$

For a simple motion model the zero velocity impact can be obtained using the relative acceleration solved from

$$0 = p_0 + v(t) \cdot (t - t_0) + \frac{a \cdot (t - t_0)^2}{2}, \tag{5}$$

using (4). Hence, one obtains $a = -\frac{v_0^2}{2p_0}$, which yield the host required acceleration

$$a_{host} = a_{obj} - a = a_{obj} - \frac{v_0^2}{2p_0}. \tag{6}$$

For constant velocity case, i.e., $a_{obj} = 0$, the risk metric simplifies further and the hypothesis test will be according to

$$\text{Prob}(-\frac{v_0^2}{2p_0} < a_{th}) > 1 - \alpha, \tag{7}$$

where a_{th} is the acceleration threshold for when we consider an accident to be imminent.

3.3 Brake System Model

To calculate the collision speed the brake system characteristics need to be known. A complete brake system model is very complex. However looking at overall vehicle

performance a simple yet accurate description can be obtained using a first order system for the acceleration. Figure 4 shows measurement data from 10 brake maneuvers with a Volvo V70 and a first order approximation The measurements were performed on warm and dry asphalt. For the simulated data a first order system was used with transfer function

$$G_{brake}(s) = \frac{k_1}{s/k_2 + 1}, \tag{8}$$

with $k_1 = 11$ and $k_2 = 7$ chosen such that the system rise time is 0.3 s and the stationary value is 11 m/s^2, Comparing traveled distance after 2 seconds we find that the difference is less than 1 m which is sufficient for our purposes. More information on typical brake system behavior can be found in [1].

4 Bayesian Estimation

Consider the state-space model

$$x_{t+1} = f(x_t, u_t, w_t), \tag{9a}$$
$$y_t = h(x_t) + e_t, \tag{9b}$$

where $x_t \in \mathbb{R}^n$ denotes the state of the system, u_t the input signal and y_t the observation at time t. The process noise w_t and measurement noise e_t are assumed independent with densities p_{w_t} and p_{e_t} respectively. Let $\mathbb{Y}_t = \{y_i\}_{i=1}^t$ be the set of observations until present time.

The Bayesian estimation problem is given by, [7],

$$p(x_{t+1}|\mathbb{Y}_t) = \int_{\mathbb{R}^n} p(x_{t+1}|x_t)p(x_t|\mathbb{Y}_t)dx_t, \tag{10a}$$
$$p(x_t|\mathbb{Y}_t) = \frac{p(y_t|x_t)p(x_t|\mathbb{Y}_{t-1})}{p(y_t|\mathbb{Y}_{t-1})}, \tag{10b}$$

where $p(x_{t+1}|\mathbb{Y}_t)$ is the prediction density and $p(x_t|\mathbb{Y}_t)$ the filtering density. The problem is in general not analytically solvable.

4.1 Particle Filter

To solve the non-tractable Bayesian estimation problem in an on-line application without using linearization or Gaussian assumptions, sequential Monte Carlo methods, or particle filters, could be used. Here only a brief description of the theory is given. For more details we refer to [3, 4, 5, 9]. The particle filter method provides an approximative Bayesian solution to (10) by approximating the probability density $p(x_t|\mathbb{Y}_t)$ by a large set of N particles $\{x_t^{(i)}\}_{i=1}^N$, where each particle has an assigned relative weight, $\gamma_t^{(i)}$, such that all weights sum to unity. The location and

weight of each particle reflect the value of the density in the region of the state space. The likelihood $p(y_t|x_t)$ is calculated from (9) yielding

$$\gamma_t = p(y_t|x_t) = p_{e_t}(y_t - h(x_t)). \tag{11}$$

By introducing a resampling step as in [5] problems with divergence can be handled. This is referred to as *sampling importance resampling* (SIR), and is summarized in Algorithm 1.

Algorithm 1 Sampling Importance Resampling (SIR)

1: Generate N samples $\{x_0^{(i)}\}_{i=1}^N$ from $p(x_0)$.
2: Compute $\gamma_t^{(i)} = p_e(y_t|x_t^{(i)})$ and normalize, i.e., $\bar{\gamma}_t^{(i)} = \gamma_t^{(i)} / \sum_{j=1}^N \gamma_t^{(j)}$, $i = 1, \ldots, N$.
3: Generate a new set $\{x_t^{(i\star)}\}_{i=1}^N$ by resampling with replacement N times from $\{x_t^{(i)}\}_{i=1}^N$, with probability $\bar{\gamma}_t^{(j)} = \text{Prob}(x_t^{(i\star)} = x_t^{(j)})$.
4: Prediction: $x_{t+1}^{(i)} = f(x_t^{(i\star)}, u_t, w_t^{(i)})$, $i = 1, \ldots, N$ using different noise realizations $w_t^{(i)}$.
5: Increase t and iterate to step 2.

The decision criterion (7) can easily be evaluated for the particle filter since the particles, $x_t^{(i)}$ reflects the location and distribution of the states. Hence, the particle filter is well suited for statistical decision making.

4.2 Extended Kalman Filter

The Bayesian recursions in Section 4 do not in general have an analytical solution. For the special case of linear dynamics, linear measurements and Gaussian noise there exist a solution, which is retrieved by the Kalman filter, [8]. For many nonlinear problems the noise assumptions are such that a linearized solution will be a good approximation. This is the idea behind the EKF, [2], where the model is linearized around the previous estimate. The time- and measurement update for the EKF are give by

$$\begin{cases} \hat{x}_{t+1|t} = f(\hat{x}_{t|t}, u_t), \\ P_{t+1|t} = F_t P_{t|t} F_t^T + G_t Q_t G_t^T, \end{cases} \tag{12a}$$

$$\begin{cases} \hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - h(\hat{x}_{t|t-1})), \\ P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}, \\ K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1}, \end{cases} \tag{12b}$$

with linearized matrices and covariances

$$F_t^T = \nabla_x f^T(x_t)|_{x_t = \hat{x}_{t|t}}, \; H_t^T = \nabla_x h^T(x_t)|_{x_t = \hat{x}_{t|t-1}}, \tag{13}$$

$$Q_t = \text{Cov}\,(w_t)\,, \; R_t = \text{Cov}\,(e_t)\,. \tag{14}$$

When a multi-modal Gaussian measurement pdf is used

$$e_t \sim \sum_{i=1}^{M} p_i \mathcal{N}(\mu_i, \sigma_i^2), \tag{15}$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian density with mean (μ) and covariance (σ^2), the measurement update equation must be modified as

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - h(\hat{x}_{t|t-1}) - \bar{m}), \tag{16}$$

where $\bar{m} = \sum_{i=1}^{M} p_i \mu_i$. If the measurement pdf is not know analytically, the approximation using a single Gaussian pdf introduces a bias (since $\bar{m} = 0$).

The decision criterion (7) is not analytically solvable when the state variables are considered Gaussian. By sampling the position and velocity distribution around the estimates a Monte Carlo integration technique can solve the decision criterion. Hence, part of the computational burden present in the particle filter is introduce in this step for the EKF method.

5 Simulations and Tests

In a simulation study we compare the traditionally EKF tracking filter with the particle filter using the previously described decision rule for braking. We also test the real-time performance in a hardware collision avoidance test platform.

5.1 Model

To simplify analysis of the system performance for different noise distributions, we consider a single scenario. The scenario to be studied is one where the target object is not moving and the vehicle with the CMbB system is approaching with constant velocity (60 km/h).

Consider the state vector with relative position p and velocity v for the x- and y-direction, where the sensor measures relative distance, azimuth and relative speed.

$$x_t = \begin{pmatrix} p_x & p_y & v_x & v_y \end{pmatrix}^T. \tag{17}$$

The tracking model with sample time T is then

$$x_{t+1} = Fx_t + Gw_t \tag{18a}$$

$$y_t = h(x_t) + e_t, \tag{18b}$$

$$F = \begin{pmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, G = \begin{pmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{pmatrix}, \tag{19}$$

$$h(x_t) = \begin{pmatrix} r \\ \varphi \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \sqrt{p_x^2 + p_y^2} \\ \arctan(p_y/p_x) \\ \frac{1}{\sqrt{p_x^2+p_y^2}}(p_x v_x + p_y v_y) \end{pmatrix}. \tag{20}$$

The process noise, w_t, and measurement noise, e_t, are assumed to be independent white noise.

5.2 Scenario

We consider two main models for the measurement sensor. In the first we use the measured range distribution from Figure 2, based on measurements collected from experimental data on a Volvo S80. We conclude that the range density is narrow and not as bimodal as expected. This is probably due to the ideal circumstances in the experiment as described in Section 2.4. We are also interested in other driving situations. Hence the second model of the measurement range noise is a bi-modal Gaussian, i.e.,

$$e_t \sim p_1 \mathcal{N}(\mu_1, \sigma_1^2) + p_2 \mathcal{N}(\mu_2, \sigma_2^2), \tag{21}$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian density with mean (μ) and covariance (σ^2). A sample of e_t belongs to one of the distributions, with probability p_i. The values are given in Table 5.2. Here $p_1 = 0.75$ and $p_2 = 0.25$ since the rear-end of the vehicle is assumed to have a larger radar cross section. The motivation for studying the bi-modal distribution is discussed in Section 2.4. Here two different parameterizations are presented to investigate the influence.

Table 1: *Results from* 1000 *Monte Carlo simulations and different cases.*

Case	**Filter**	**Range pdf** p_{e_t}	**RMSE** (pos)[m]	**RMSE** (vel)[m/s]	**Coll.vel** [m/s]	σ **Coll.vel** [m/s]
I	SIR	$0.75\mathcal{N}(0, 0.4^2) + 0.25\mathcal{N}(1.6, 0.4^2)$	0.11	0.29	-6.59	0.51
I	EKF	$0.75\mathcal{N}(0, 0.4^2) + 0.25\mathcal{N}(1.6, 0.4^2)$	0.16	0.29	-6.61	0.53
II	SIR	$0.75\mathcal{N}(0, 0.2^2) + 0.25\mathcal{N}(1, 0.4^2)$	0.08	0.29	-6.56	0.49
II	EKF	$0.75\mathcal{N}(0, 0.2^2) + 0.25\mathcal{N}(1, 0.4^2)$	0.11	0.29	-6.56	0.51
III	SIR	Empirical	0.06	0.29	-6.52	0.47
III	EKF	Empirical	0.06	0.29	-6.52	0.47

The range rate and azimuth were found to be well approximated by Gaussian distributions, with $\sigma_{\dot{r}} = 0.2$ and $\sigma_{\varphi} = 0.01$.

We use a longitudinal acceleration process noise, $Q \approx 0.5/T$, to model driving behavior and model imperfections.

A simple brake system model according to (8) is used in the simulations, with parameters k_1 and k_2 chosen to give a brake system rise-time of 300 ms and a maximum deceleration of 9.8 m/s^2. The somewhat lower maximum deceleration

(compared to section 2) is due to the fact that the test track exhibit optimal frictional conditions.

The brake decision is based on a hypothesis test on the expected required acceleration according to (7). We have chosen the acceleration threshold to $a_{th} = -8$ m/s^2 and the confidence level to $\alpha = 0.05$. For the SIR approach the hypothesis is evaluated for each particle.

We perform 1000 *Monte Carlo* (MC) simulations for both SIR and EKF. In the particle filter we use $N = 5000$ particles and the same amount of samples in the stochastic integration for the EKF.

5.3 Simulation Results

The different simulations from the Monte Carlo study are summarized in Table 5.2 together with tracking and CMbB system performance. Three different cases were considered. In case I and II two different bi-modal Gaussian range distributions were used and in case III the empirical range distribution from Figure 2 were used. All other values were the same for all the simulation cases. For non-Gaussian range distributions such as the bimodal-Gaussian proposed for the measurement noise, the SIR method increases estimation performance (position RMSE) slightly. The mean difference in collision speed is basically not affected, however the EKF has a somewhat larger collision speed variance, but probably insignificant. The result is under the assumptions of quite accurate range rate and azimuth measurements and a high measurement update rate, $f = 1/T = 20$ Hz. Hence, for systems that do not measure the relative speed more significant differences are expected.

5.4 Hardware and Real-Time Issues

In the simulations presented in Section 5.1 the entire algorithm was implemented in standard MATLAB code. However, the ultimate goal with the particle filter based approach is to incorporate the algorithm and test the idea in an on-line application in a collision avoidance system. In Figure 5 the test system is shown mounted inside the test vehicle. The test platform uses dSpace hardware (equipped with a 1 GHz power PC) together with Simulink and *Real-Time Workshop* (RTW). Hence, the algorithm must be written in C-code and MATLAB/MEX functions for Simulink. The entire environment is then compiled using the RTW-compiler to the dedicated hardware platform. In Figure 6 the Simulink diagram is shown for the underlying MEX C-code functions, i.e., the particle filter and the decision making algorithm. These are incorporated in the testbed using the RTW-compiler. The entire algorithm runs faster then real-time on the dedicated hardware, with $T = 0.05$ s and $N = 5000$ particles.

6 Conclusions

We have implemented a decision rule in a CMbB system for late braking using a hypothesis test based on estimates of the relative longitudinal dynamics. Both

Figure 5: *The collision avoidance hardware in the test vehicle.*

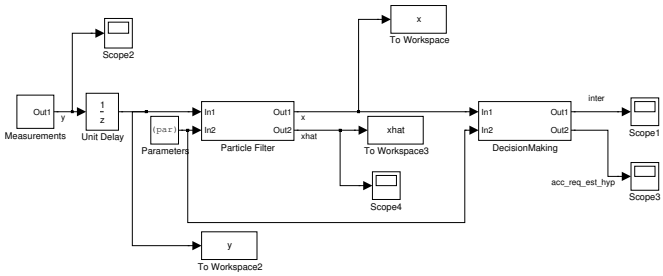


Figure 6: MATLAB *Simulink diagram of the CMbBS model used to generate RTW-code for the dedicated hardware.*

an EKF and a particle filter were evaluated for different noise assumptions. For non-Gaussian range distributions such as the bimodal-Gaussian proposed for the measurement noise, the SIR method increases estimation performance slightly. However, the range error is in the order of decimeters for all tested scenarios, so the mean difference in collision speed is basically not affected. The EKF has a somewhat larger collision speed variance, but probably insignificant. The result is under the assumption of quite accurate range rate and azimuth measurements, and a moderate process noise. Hence, a greater difference might occur if for instance the range rate is not measured or if more maneuverability, i.e., larger process noise is considered.

The computational cost of the EKF method is somewhat smaller than for the

SIR, but rather close. This is due to the fact that the EKF uses a stochastic integration to calculate the probability of collision. So for a decision rule like (2) a particle filter approach might be preferable. It should also be kept in mind that the scenario studied here is very simple (from a tracking sense), for more complex scenarios with maneuvering target and tracking platform the difference between the methods may be larger. Also the particle filter method is more flexible, so if a more complex state-space model is used, linearization errors in the dynamic model can be avoided.

In the simulation study the simulations for both the SIR and EKF approach were run faster than real-time on an ordinary desk-top computer. The particle filter was also implemented in the Volvo testbed hardware (used in vehicle tests) using the MATLAB Simulink RTW compiler. The tracking filter and decision algorithm executed faster than the real-time constraint, $T = 0.05$, using $N = 5000$ particles for the tracking model described earlier.

Acknowledgment

The authors would like to thank Volvo Car, Sweden, especially Fredrik Lundholm and Lena Westervall for providing measurement data and a hardware platform to test the algorithm on.

References

[1] U. Adler. *Automotive Handbook*. Number ISBN: 0-7680-0669-4. Robert Bosch GmbH, 5th edition, 2000.

[2] B.D.O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, NJ, 1979.

[3] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Linköping Studies in Science and Technology. Dissertations No. 579, Linköping University, Linköping, Sweden, 1999.

[4] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[5] N. J. Gordon, D. J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[6] J. Jansson. *Tracking and Decision Making for Automotive Collision Avoidance*. Linköping Studies in Science and Technology. Licentiate Thesis No. 965, Linköping University, Linköping, Sweden, September 2002.

[7] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, 1970.

[8] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the AMSE–Journal of Basic Engineering*, 82:35–45, 1960.

[9] R. Karlsson. *Simulation Based Methods for Target Tracking.* Linköping Studies in Science and Technology. Licentiate Thesis No. 930, Linköping University, Linköping, Sweden, February 2002.

[10] L. Ljung. *System Identification Toolbox.* The MathWorks, Inc.

[11] L. Ljung. *System Identification, Theory for the User.* Prentice Hall, Englewood Cliffs, New Jersey, second edition, 1999.

[12] J. R. Treat, N. S. McDonald, D. Shinar, R. D. Hume, R. E. Mayer, R. L. Stansifer, and N. J. Castellan. Tri-level study of the causes of traffic accidents. Technical report, NHTSA, 1977.

[13] B. Zhu. Potential effects on accidents from forward collision warning/avoidance system. Master's Thesis LiTH-ITN-EX-150-SE, Linköping University, Linköping, Sweden, 2001.

PhD Dissertations, Division of Automatic Control, Linköping University

M. Millnert: Identification and control of systems subject to abrupt changes. Thesis no. 82, 1982. ISBN 91-7372-542-0.

A.J.M. van Overbeek: On-line structure selection for the identification of multivariable systems. Thesis no. 86, 1982. ISBN 91-7372-586-2.

B. Bengtsson: On some control problems for queues. Thesis no. 87, 1982. ISBN 91-7372-593-5.

S. Ljung: Fast algorithms for integral equations and least squares identification problems. Thesis no. 93, 1983. ISBN 91-7372-641-9.

H. Jonson: A Newton method for solving non-linear optimal control problems with general constraints. Thesis no. 104, 1983. ISBN 91-7372-718-0.

E. Trulsson: Adaptive control based on explicit criterion minimization. Thesis no. 106, 1983. ISBN 91-7372-728-8.

K. Nordström: Uncertainty, robustness and sensitivity reduction in the design of single input control systems. Thesis no. 162, 1987. ISBN 91-7870-170-8.

B. Wahlberg: On the identification and approximation of linear systems. Thesis no. 163, 1987. ISBN 91-7870-175-9.

S. Gunnarsson: Frequency domain aspects of modeling and control in adaptive systems. Thesis no. 194, 1988. ISBN 91-7870-380-8.

A. Isaksson: On system identification in one and two dimensions with signal processing applications. Thesis no. 196, 1988. ISBN 91-7870-383-2.

M. Viberg: Subspace fitting concepts in sensor array processing. Thesis no. 217, 1989. ISBN 91-7870-529-0.

K. Forsman: Constructive commutative algebra in nonlinear control theory. Thesis no. 261, 1991. ISBN 91-7870-827-3.

F. Gustafsson: Estimation of discrete parameters in linear systems. Thesis no. 271, 1992. ISBN 91-7870-876-1.

P. Nagy: Tools for knowledge-based signal processing with applications to system identification. Thesis no. 280, 1992. ISBN 91-7870-962-8.

T. Svensson: Mathematical tools and software for analysis and design of nonlinear control systems. Thesis no. 285, 1992. ISBN 91-7870-989-X.

S. Andersson: On dimension reduction in sensor array signal processing. Thesis no. 290, 1992. ISBN 91-7871-015-4.

H. Hjalmarsson: Aspects on incomplete modeling in system identification. Thesis no. 298, 1993. ISBN 91-7871-070-7.

I. Klein: Automatic synthesis of sequential control schemes. Thesis no. 305, 1993. ISBN 91-7871-090-1.

J.-E. Strömberg: A mode switching modelling philosophy. Thesis no. 353, 1994. ISBN 91-7871-430-3.

K. Wang Chen: Transformation and symbolic calculations in filtering and control. Thesis no. 361, 1994. ISBN 91-7871-467-2.

T. McKelvey: Identification of state-space models from time and frequency data. Thesis no. 380, 1995. ISBN 91-7871-531-8.

J. Sjöberg: Non-linear system identification with neural networks. Thesis no. 381, 1995. ISBN 91-7871-534-2.

R. Germundsson: Symbolic systems – theory, computation and applications. Thesis no. 389, 1995. ISBN 91-7871-578-4.

P. Pucar: Modeling and segmentation using multiple models. Thesis no. 405, 1995. ISBN 91-7871-627-6.

H. Fortell: Algebraic approaches to normal forms and zero dynamics. Thesis no. 407, 1995. ISBN 91-7871-629-2.

A. Helmersson: Methods for robust gain scheduling. Thesis no. 406, 1995. ISBN 91-7871-628-4.

P. Lindskog: Methods, algorithms and tools for system identification based on prior knowledge. Thesis no. 436, 1996. ISBN 91-7871-424-8.

J. Gunnarsson: Symbolic methods and tools for discrete event dynamic systems. Thesis no. 477, 1997. ISBN 91-7871-917-8.

M. Jirstrand: Constructive methods for inequality constraints in control. Thesis no. 527, 1998. ISBN 91-7219-187-2.

U. Forssell: Closed-loop identification: Methods, theory, and applications. Thesis no. 566, 1999. ISBN 91-7219-432-4.

A. Stenman: Model on demand: Algorithms, analysis and applications. Thesis no. 571, 1999. ISBN 91-7219-450-2.

N. Bergman: Recursive Bayesian estimation: Navigation and tracking applications. Thesis no. 579, 1999. ISBN 91-7219-473-1.

K. Edström: Switched bond graphs: Simulation and analysis. Thesis no. 586, 1999. ISBN 91-7219-493-6.

M. Larsson: Behavioral and structural model based approaches to discrete diagnosis. Thesis no. 608, 1999. ISBN 91-7219-615-5.

F. Gunnarsson: Power control in cellular radio systems: Analysis, design and estimation. Thesis no. 623, 2000. ISBN 91-7219-689-0.

V. Einarsson: Model checking methods for mode switching systems. Thesis no. 652, 2000. ISBN 91-7219-836-2.

M. Norrlöf: Iterative learning control: Analysis, design, and experiments. Thesis no. 653, 2000. ISBN 91-7219-837-0.

F. Tjärnström: Variance expressions and model reduction in system identification. Thesis no. 730, 2002. ISBN 91-7373-253-2.

J. Löfberg: Minimax approaches to robust model predictive control. Thesis no. 812, 2003. ISBN 91-7373-622-8.

J. Roll: Local and piecewise affine approaches to system identification. Thesis no. 802, 2003. ISBN 91-7373-608-2.

J. Elbornsson: Analysis, estimation and compensation of mismatch effects in A/D converters. Thesis no. 811, 2003. ISBN 91-7373-621-X.

O. Härkegård: Backstepping and control allocation with applications to flight control. Thesis no. 820, 2003. ISBN 91-7373-647-3.

R. Wallin: Optimization algorithms for system analysis and identification. Thesis no. 919, 2004. ISBN 91-85297-19-4.

D. Lindgren: Projection methods for classification and identification. Thesis no. 915, 2005. ISBN 91-85297-06-2.