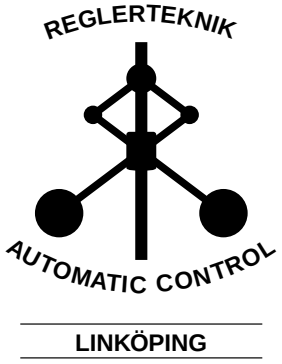
Linköping studies in science and technology. Licentiate Thesis.
No. 1611

On Calibration of Ground Sensor Networks

Marek Syldatk



Division of Automatic Control
Department of Electrical Engineering
Linköping University, SE-581 83 Linköping, Sweden
http://www.control.isy.liu.se
marek@isy.liu.se

Linköping 2013

This is a Swedish Licentiate's Thesis.

Swedish postgraduate education leads to a Doctor's degree and/or a Licentiate's degree.
A Doctor's Degree comprises 240 ECTS credits (4 years of full-time studies).
A Licentiate's degree comprises 120 ECTS credits,
of which at least 60 ECTS credits constitute a Licentiate's thesis.

Linköping studies in science and technology. Licentiate Thesis.
No. 1611

On Calibration of Ground Sensor Networks

Marek Syldatk

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

ISBN 978-91-7519-540-7 ISSN 0280-7971 LIU-TEK-LIC-2013:45

Printed by LiU-Tryck, Linköping, Sweden 2013

To my Parents!

Abstract

Sensor networks are everywhere around us. Developments in sensor technology and advances in hardware miniaturization open up brand-new application areas. In the future networks of cheap and small sensor nodes will be deployed for a variety of purposes. Military needs have been a major motivation for the development in the past, but today it has changed. Other applications such as traffic monitoring, security threat detection, ecology and environmental protection are the new driving forces behind further development.

The thesis considers the problem of calibration of ground sensor networks. In order to perform its operational tasks – detection, classification and tracking of objects of interest, the network has to be correctly calibrated. Improper calibration might result in a degraded performance, problems with data association and appearance of multiple track instances representing one object.

In order to find the unknown calibration parameters (biases), in most cases we need to use reference targets with known positions. If such targets are not available, one has to use opportunistic targets and simultaneously estimate both target positions and bias parameters. In this thesis, the expectation maximization algorithm is applied to that problem, where the unknown states are treated as latent (unknown) variables in the process of bias estimation.

Next, the problem of estimating a large number of calibration parameters is tackled. In the case when the measurement data is not informative enough – due to a limited range of sensors or a small number of samples – standard approaches such as the least squares algorithm might provide unreliable results. One solution to the problem is to apply a regularization (or prior in a Bayesian case). In this thesis, the problem of selecting the parameters (the so called hyper-parameters) for the regularization process, based on the set of measurements, is considered. The solution is provided through the evidence approximation method, where both the bias parameters and the hyper-parameters are estimated simultaneously. As a result, one obtains a robust algorithm that, thanks to the application of Occam's razor, allows to find the good trade-off between model complexity and its fit to the data.

Finally, both methods are combined together, in order to provide a robust and accurate algorithm for the calibration of sensor networks using targets of opportunity.

The applicability of algorithms was also verified during field trials with good final outcome, confirming the expected performance.

Populärvetenskaplig sammanfattning

Sensornätverk finns överallt omkring oss. Klassiska exempel är meteorologiska stationer för väderprognoser samt seismometrar för att lokalisera jordbävningar och explosioner. Utvecklingen av små och billiga sensorer möjliggör sensornätverk i en helt ny omfattning jämfört med tidigare. En vanlig vision är att sprida ut tusentals små enheter som decentraliserat samverkar för att lösa komplexa problem. Ett specifikt exempel på en sådan vision är att alla världens smarta telefoner utbyter sensordata med varandra. I princip kan accelerometrarna användas för att upptäcka jordbävningar, mikrofonerna kan lokalisera bullerkällor och explosioner, GPS-mottagarna kan modellera atmosfäriska fenomen, barometrar och termometrar kan matas in i väderprognosmodeller, dopplereffekter i mottagna radiosignaler kan användas för att följa alla jordens fordon, etc. Algoritmer för detta finns redan idag.

En flaskhals i exemplet ovan och rent allmänt i utrullning av sensornätverk är att sensorerna måste vara väl kalibrerade samt att deras positioner måste vara kända för att meningsfulla slutsatser ska kunna dras. Denna avhandling behandlar detta problem i detalj. Idag krävs tidsödande manuellt arbete för att mäta ut exakt var sensorerna placeras samt att jämföra deras så kallade biasparametrar. Båda problemen kan lösas halvautomatiskt, t. ex. för ett mikrofonnätverk genom att flytta runt en ljudkälla till kända positioner. Denna avhandling beskriver en metod som gör detta helautomatiskt genom att utnyttja tillfälliga ljudkällor som råkar passera förbi, t. ex. en bil eller fågel. Metoden bygger på att alla kalibreringsparametrar skattas simultant med ljudkällans position.

De sensorer som använts i detta arbete har mellan tre och tio kalibreringsparametrar, så för nätverk med hundratals eller kanske tusentals sensorer så kan skattningarna lätt bli dåliga. Avhandlingen beskriver en utvidgning av en grundmetod som bygger på att de flesta parametrar oftast är tillräckligt välkalibrerade, och detekterar automatiskt vilka parametrar som behöver kalibreras.

Metoderna är utvärderade på data från fältförsök med akustiska sensornätverk med utmärkta resultat.

Acknowledgments

First of all I would like to thank my supervisor Prof. Fredrik Gustafsson for the excellent guidance and research support leading to this thesis. My former co-supervisors also deserve a special thank: Dr. Egils Sviestins, for a supervisory during my stay at SAAB, and Dr. Umut Orguner to whom I owe plenty of research ideas.

Being a member of the Automatic Control group at Linköping University is a great honor. Thus I would like to thank Prof. Svante Gunnarsson, Prof. Lennart Ljung and Prof. Fredrik Gustafsson for the great leadership and invitation to join the group. The division coordinator Ninna Stensgård deserves my deepest gratitude for all the help with administrative tasks.

My adventure with postgraduate studies abroad would not happen without a suggestion from Prof. Zdzisław Kowalczuk, who was my supervisor during my Master's degree studies at Gdańsk University of Technology. He is the one who gave me the idea to take the challenge and apply for the position.

Before moving to Linköping I spent two years doing research at SAAB in Järfälla. I hereby would really like to thank Dr. Sviestins and Dr. Mats Ekman for giving me the opportunity and being great supervisors. The Data Fusion group was a great place to work, especially thanks to people there – I will always remember the superb time I had during the formal and the informal after works and interesting lunchtime discussions. Here I also need to mention Hans-Roland Gottfridsson and thank him for all the intellectual discussions!

Both positions, at SAAB and at Linköping University were part of the MC IMPULSE project. The research would not succeed without the funding from the EU's 7^{th} Framework Program, under the grant agreement number 238710.

The MC IMPULSE is not only about research. It is also a magnificent group of sharp researchers and supervisors – without them all the Summer Schools and conferences would not be the same. I would especially like to thank Fotios Katsilieris and Edson *"The Crazy Brazilian Ninja Scientist"* Hiroshi Aoki (for a superb time in Chicago, Singapore and during the exploration of Bali), Dr. Nikolay Petrov (for late night adventures in Manchester and Singapore) as well as Manon Kok and Michael Roth (who are also my LiU colleagues). I am also thankful to Dr. Murat Aslan for fruitful technical discussions during our stay at SAAB.

The number of errors in this thesis would be much higher without a Sisyphean labor of the proof-reading team: Dr. Gustaf Hendeby, Dr. Saikat Saha, Lic. Zoran Sjanic, Lic. André Carvalho Bittencourt, Hanna Nyqvist and Manon Kok. I would also like to thank Prof. Gustafsson for a remarkable amount of time spent *(un)-Polish-ing* my publications before submission. The thesis would not have its shape without Dr. Henrik Tidefelt and Dr. Hendeby, who have constructed the LaTeX class for the layout. Thank you all!

Becoming a part or the Automatic Control group at LiU is a remarkable experience by itself. It is one of these groups that makes you feel being in the right place from the beginning. All the fika breaks and Friday after-works make it a great place to work, to do research and to study.

Lic. *Chef* Sina Khoshfetrat Pakazad deserves a special thanks for all his delicious winter (and summer) Halloumi BBQs. Together with André, Jonas Linder, Manon and Zoran and all the others – thank you guys for being great friends and great companions during the hiking adventures, down the Ågatan, from bar to bar. Niklas also deserves a special badge for being able to handle my persona in the same office for over a year! As well as André and his girlfriend Alicia, my great flatmates. Thank you guys and thank you to all I forgot to mention by name.

Last but not least, I would like to thank my Polish friends and my family. I always have a great time meeting you when I am back home. These visits keep me sane and strongly motivated to work hard when I go back to the other side of the Baltic Sea. Most of all though I would like to thank my Parents, for their love and tremendous support in my decision to move abroad. Without you I would not be here where I am now!

Na wstępie chciałbym podziękować Panu Prof. Zdzisławowi Kowalczukowi z Politechniki Gdańskiej, który to zasugerował mi podjęcie tego wyzwania i wyjazd na studia doktoranckie za granicę.

Kolejne wielkie "Dziękuję!" należy się wszystkim moim bliskim przyjaciołom i całej mojej rodzinie, bo to dzięki Wam zawsze mam do czego wracać.

Najważniejsze podziękowania należą się jednak moim Rodzicom, bo to dzięki Waszemu wsparciu, na które zawsze mogę liczyć, udało mi się dotrzeć tam gdzie w tej chwili jestem. Dziekuję!

Linköping, August 2013
Marek Syldatk

Contents

II Publications

Notation

Used notation

Notation	Meaning
$k = 1 \dots K$	Time step
x_k	Target state vector at k-th time step
y_k	Sensor measurement vector at k-th time step
$X_k = \{x_i\}_{i=1}^{k}$	Set of target states from x_1 to (current) x_k
$Y_k = \{y_i\}_{i=1}^{k}$	Set of sensor measurements from y_1 to (current) y_k
θ	Bias vector
λ	Regularization parameter
α	Hyper-parameter
η_k	Process noise at k-th time step
ν_k	Measurement noise at k-th time step
P	State covariance matrix
Q	Process noise covariance matrix
R	Measurement noise covariance matrix

Abbreviations

Abbreviation	Meaning
BLUE	Best Linear Unbiased Estimator
CRLB	Cramér-Rao Lower Bound
CV	Constant Velocity
DOA	Direction of Arrival
EA	Evidence Approximation
EKF	Extended Kalman Filter
EM	Expectation Maximization
FIM	Fisher Information Matrix
GPS	Global Positioning System
GSN	Ground Sensor Network
HMM	Hidden Markov Model
KF	Kalman Filter
LS	Least Squares
MAP	Maximum a Posteriori
MEMS	Micro Electro-Mechanical Systems
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
NLS	Nonlinear Least Squares
NWLS	Nonlinear Weighted Least Squares
POI	Point of Interest
PF	Particle Filter
PMF	Point Mass Filter
RMS	Root Mean Square
RSS	Received Signal Strength
SN	Sensor Network
TDOA	Time Difference of Arrival
TOA	Time of Arrival
UKF	Unscented Kalman Filter
WLS	Weighted Least Squares

Part I

Background

1 Introduction

This introductory chapter presents a brief background and motivation behind the work presented in this thesis. An introduction to sensor networks and the calibration problem is provided. Published works of relevance, in which the author has been involved in, are listed together with corresponding contributions. Last section concludes the chapter and gives an outline to the rest of the thesis.

1.1 Background and Motivation

Sensor networks (SN) are constantly becoming more present in our everyday life. In recent years, a huge amount of research in this area has lead to a great improvement in technology and algorithms used in the development of sensor networks. Significant advances in digital electronics, wireless communication and micro electro-mechanical systems (MEMS) have made deployment of sensors networks relatively cheap and more available.

Initially used in the military area, in the last 20 years SN have started to play a more significant role in areas like health monitoring, terrorist threat detection, environment monitoring, traffic monitoring and many more. Nowadays, a great amount of research is focused in the area of unattended ground sensor networks (GSN) that can be quickly deployed e.g. in the battlefield. To make it possible, the networks need to have a flexible structure, be energy efficient and have an ability to self-organize in order to provide reliable observations.

Progress in the technology led us from primitive centralized networks initially to the most sophisticated unattended wireless SN today. Networks, originally designed to monitor a certain phenomenon (usually a ground target), have evolved into what we can undoubtedly call intelligent networks. The principles underly-

ing their operational purpose remained the same though: to detect, classify and track objects moving within the area covered by the network.

In order to perform the task of correct detection, classification and tracking of the targets, a precise calibration and knowledge of the sensor positions within the network is crucial. Any disturbances (called bias parameters, biases) in the network might result in degraded performance of the GSN, appearance of ghost tracks (indication of inexistent targets) or wrong measurement-to-track association in a multi-target scenario.

In this thesis two subsequent problems regarding GSN calibration are considered: calibration using targets of opportunity (as a reference) and a problem of sparse estimation of a bias vector.

1.1.1 Calibration using Targets of Opportunity

The correct calibration of the network is a crucial but not a trivial problem. Standard approaches might suggest the use of stationary beacons or vehicles of known trajectories as a calibration reference. Measurements can be collected using for example a global positioning system (GPS) and then simple estimation algorithms can be applied to obtain the bias parameters. In cases where the GPS is unavailable, or reference targets cannot be deployed, it is necessary to use targets of opportunity to calibrate the network.

There are many approaches in the literature for handling the problem of simultaneous target state and bias estimation. One of the basic ideas is to augment the state vector with the unknown parameters and then use one of the well known filtering methods like Kalman filter (KF). Since this may lead to computational problems, approaches to overcome these difficulties by decoupling the problem into two sub-tasks have been proposed, both in online and offline applications (Vermaak et al. (2005), Ignagni (1981), Sviestins (1999)) .

In the research leading to the contributions presented in this thesis, an offline approach to the problem was considered. Since for the GSN the amount of available measurements is usually low, causing problems with observability, one is forced to use any (available) extra knowledge about the scenario or bias parameter vector when solving the problem. Additional information can be extracted by applying constraints to the model or by utilizing knowledge of the structure of the bias vector.

To tackle that problem an expectation maximization (EM) algorithm is applied, that provides a solution to the maximum likelihood estimation in a presence of the unknown (latent) variables. The method is extended with some enhancements improving the quality of both the state and the bias estimation.

1.1.2 Sparse Estimation

Another problem can be a large number of unknown parameters to be estimated. It might happen that some of them might be unobservable. The problem appears particularly relevant in scenarios, where the amount of information contained in

the measurements is low. It might be a result of a limited range of the sensors or a very few observations. To handle that problem one has to consider applying a regularization to the estimation process, corresponding to the use of priors in a Bayesian framework. The problem arises when the regularization parameters cannot be set arbitrarily.

In parallel it is also reasonable to consider a reduction of the bias model complexity, since it is usually not necessary to estimate the full set of parameters. Under the assumption that a large number of parameters can be dropped (set to zero due to their irrelevance – this feature is called sparsity), the problem of automatic selection of regularization/prior parameters is addressed in the thesis.

To solve the problem, a Bayesian concept of an evidence function is applied. Regularization parameters are computed simultaneously with bias parameters, based on the set of collected measurements. It is shown that this method applies Occam's razor, that leads to an optimal balance between the model complexity (number of parameters set to zero) and fit to the measurement data. The sparsity feature is implied through the shrinkage priors.

1.2 Contributions

Published works of relevance to this thesis are listed below in chronological order, together with short summary and corresponding contributions.

Paper A: Expectation Maximization Algorithm for Calibration of Ground Sensor Networks using a Road Constrained Particle Filter

> M. Syldatk, E. Sviestins, and F. Gustafsson. Expectation maximization algorithm for calibration of ground sensor networks using a road constrained particle filter. In *Proceedings of the 15th International Conference on Information Fusion (FUSION)*, pages 771–778, Singapore, July 2012.

Summary: The research presented in this paper was conducted during my two year stay at Saab AB in Järfälla, Sweden between June 2010 and May 2012. The problem of calibrating the acoustic sensor network with only one target of opportunity and limited number of observations is considered. In order to improve the accuracy of the estimate an expectation maximization algorithm is applied, using a road constrained particle filter/smoother to provide necessary state estimates. The EM algorithm is used to obtain a maximum a posteriori (MAP) estimates of the bias parameters.

Contribution: The main contribution is the derivation of a road constrained particle filter and smoother to be applied in the EM framework. Application of the road constraints minimizes the degrees of freedom and results in much better target state estimates, implying better calibration results.

Paper B: Simultaneous Tracking and Sparse Calibration in Ground Sensor Networks using Evidence Approximation

> M. Syldatk and F. Gustafsson. Simultaneous tracking and sparse calibration in ground sensor networks using evidence approximation. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

Summary: This paper tackles the problem of applying regularization to the process of sensor network calibration with targets of opportunity. Application of the prior reduces problems related to low observability. The parameters of the prior are determined using the two-stage Bayesian evidence approximation (EA) framework, where through the evidence function one implies application of Occam's razor. This feature allows for a good balance between model complexity (number of non-zero parameters) and fit to the data. As an extra feature the algorithm has the ability to detect non-zero bias parameters and determine which parameters are well determined by measurement data.

Contribution: The main contribution is an application of the EA framework to the calibration of sensor network. The algorithm is also modified to handle the case of an unknown set of target states. Good estimation results were obtained compared to the standard maximum likelihood (ML) approach, especially with high sparsity in the bias vector.

Paper C: Expectation Maximization Algorithm for Simultaneous Tracking and Sparse Calibration

> M. Syldatk and F. Gustafsson. Expectation maximization algorithm for simultaneous tracking and sparse calibration of sensor networks. In *IEEE Transactions on Signal Processing (TSP)*. To be submitted.

Summary: In the paper the problem of applying EA to the calibration of sensor networks is exploited again. Targets of opportunity are used to obtain a MAP estimate in the expectation maximization framework. Combining the evidence and expectation maximization framework results in a robust and accurate algorithm for sparse parameter estimation.

Contribution: This paper combines together results from the previous two papers. A two stage inner-looped algorithm is derived, combining features of the EM algorithm (handling the ML case with latent variables) and the EA algorithm (application of the prior and evaluation of sparsity through Occam's razor).

1.3 Thesis Outline

After the introduction to the problem and basic contributions in Chapter 1, a more detailed description of GSN is presented in Chapter 2. Starting from the basic description of SN, through a general sensor measurement model to the introduction of different sensor types at the end. In Chapter 3 the calibration problem is presented in detail. The description is supported by the analysis of the

effects incorrect calibration might have on SN performance, together with bias sensitivity analysis. Next, in Chapter 4 least squares (LS) techniques are presented together with maximum likelihood and regularization methods. Later basic state estimation techniques are presented together with analysis of parameter observability and estimation performance. Chapter 5 describes more sophisticated methods for simultaneous state and bias parameter estimation, including expectation maximization and evidence approximation algorithms. Finally, the thesis is summarized in Chapter 6, with overall conclusions and suggestions for future research.

2

Sensor Networks

Over the last few decades the area of ground sensor networks has grown from being in an experimental phase into a highly developed and sophisticated technology. A progress in development of different types of sensors, including miniaturization, communication methods and power consumption performance has resulted in wider and wider application of sensor networks in everyday life.

The chapter provides a short description and a classification of the typical sensor network. A general mathematical model for the sensor measurements is provided, followed by a detailed characterization of different types of sensors used in the area of ground sensor networks. The material presented in this chapter is based on the extensive review of sensor networks by Chong and Kumar (2003) and Akyildiz et al. (2002), where a great survey on wireless sensor networks can be found. The overview of sensor models is based on the theory provided in Gustafsson (2012).

2.1 Sensor Network Definition

The objective of this section is to provide a formal definition of SN together with examples of sensor networks applications. A more detailed description of a typical sensor node and the way it operates within the network is given. The description is followed by an overview of the different network configurations (centralized, decentralized and distributed) and the way data fusion is performed. At the end of the section a generic mathematical model for a sensor is given.

A sensor network can be understood as a set of nodes, which usually contain sensors. Nodes communicate with a central node or with each other, using wired or wireless connection, and are used to observe a certain phenomenon. Methods

presented in this thesis are mainly related to the area of ground sensor networks. In the definition of a ground sensor network (GSN) we include all networks with nodes placed on the ground and distributed over a certain area to be surveilled.

There is a variety of applications where sensor networks can be used. The most common for the GSN is surveillance, where a number of targets are observed. Current and possible applications include military sensing, security provision, environmental monitoring, traffic monitoring and many more. In this thesis, the area of ground surveillance will be covered, where the targets to be considered are for example road vehicles, airplanes or people. For that type of applications, especially in the military area, fast deployment, performance and reliability of the network are necessary to perform the main operational tasks. Sensor networks in that case are usually a part of *command*, *control*, *communications*, *intelligence*, *surveillance*, *target acquisition and reconnaissance* (C3ISTAR) systems.

A typical network node contains four basic components: a sensing unit, a processing unit, a communication interface and a power unit (Akyildiz et al., 2002). Different types of sensors can be classified according to type of measurements, range, size, communication method etc. Based on the type of measurements gathered, we can distinguish between the basic sensor types used in GSN, as: *acoustic sensors*, *seismic sensors*, *radar sensors*, *light sensors*, *cameras*, *magnetometers* and many more. Further in this chapter, in Section 2.3, some of the aforementioned sensor types will be described in detail.

Depending on the configuration of the network we can distinguish centralized, decentralized and distributed networks (Figure 2.1). In the first type all sensors are gathering information and then deliver it to a central node, where the fusion is done to get an overview of the situation. On the other hand, in decentralized networks, fusion is performed at selected independent nodes. The third type of SN does not have a strict structure, and thus each sensor can be a fusion node. This of course implies a problem of determining which sensors, among those observing the target at the moment, should perform the fusion. The ability to self-organize is crucial in order to improve the performance, limit the bandwidth and thus the power consumption, which is especially important for large battery powered networks.

The goal of a ground sensor network is to detect, classify and track object within the area of the network. One way it can be achieved will be presented on the example of a distributed sensor network.

In that type of SN each sensor has an ability to autonomously detect the target. After the detection of sound (for acoustic sensors), vibrations (for seismic sensors), change in magnetic field (for magnetometers) or visual image (for cameras) has been indicated, the signal can be used to perform a (single-node) target classification (Brännström et al., 2004). This process can be understood as determining the type of the target (person, road vehicle of plane).

Sometimes the term classification is used to describe the process of finding differences between objects of the same type, e.g. determining the model of the car

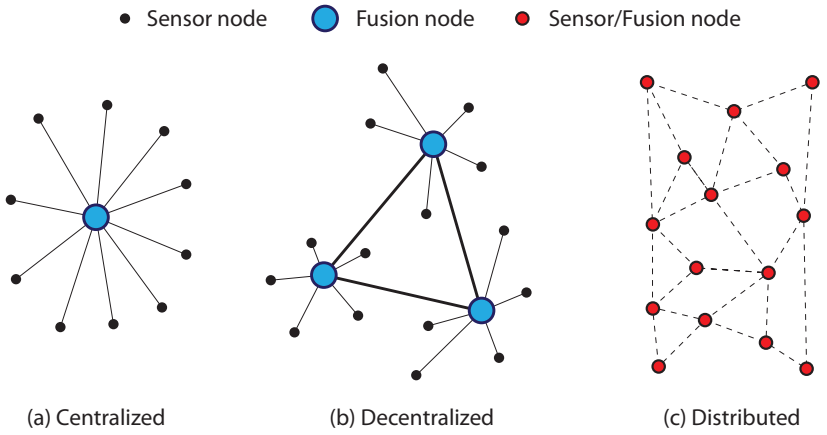


Figure 2.1: *Three different types of sensor network configurations: centralized, decentralized and distributed.*

or distinguishing two discussing people. For each target there usually exists a special signature, obtained by feature extraction, that can be computed based on the measurements. It can be further compared with a database of signatures, in order to determine the target class.

It is common that a single node has a limited ability to perform tracking alone, thus sensor nodes must cooperate with each other. In order to correctly fuse the data from multiple sensors, a correct association needs to be done – it has to be determined which groups of sensors observe the same target. In parallel, groups of sensors can be combined in order to improve the quality of sensing and classification. A good example is an acoustic array, where measurements from three or more sensors (in that case microphones) are processed together. Thanks to that, an angle (bearing) to the target can be determined. Further, by using two or more arrays to perform triangulation, an exact position of the target can be determined.

After performing the classification/data association and making sure the sensors observe the same target, a track has to be initiated. Usually the sensor node closest to the observed target initiates the track – it becomes a track agent. When the track is initiated, all neighboring sensors send their observations to the processing node (track agent), so that it can be fused into one consistent scenario. Unless a new target is observed, none of the other sensors are allowed to initiate new tracks. Also, in this step the classification can be improved by combining data from multiple sensors.

Track agents move across the network, following the observed targets that are present within the observed area. It is their task to manage the sensors currently observing the target, to coordinate the communication between nodes and between the track agents and the operators (users) as well as perform the fusion.

There of course appear a number of issues concerning the network organization in that case. A curious reader is encouraged to take a look at the paper by Brännström et al. (2004), where problems of automatic data association, sensor selection, data fusion, information lookup and finally robustness and scalability are further discussed.

In the contrary to the distributed SN, the problem of network self-organization does not apply to the centralized and decentralized networks. In that configurations all sensors are connected to, and managed by, one central node (for centralized SN) or few fusion nodes (also called routers; for decentralized SN). These central nodes are then responsible for the fusion and the classification after sensing nodes have detected the target. An example of a decentralized ground sensor network is presented in Figure 2.2.

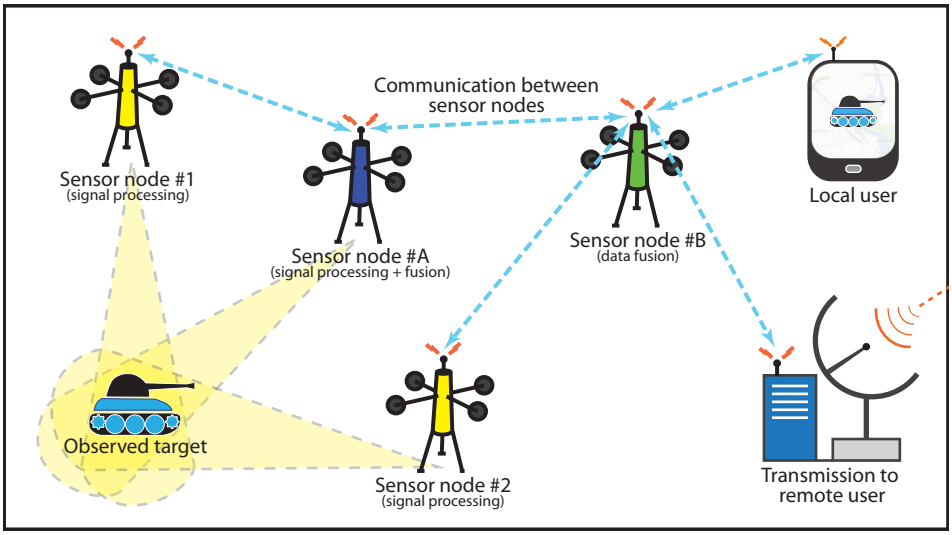


Figure 2.2: *An example of a ground sensor network (GSN). Sensors #1, #2 and #A observe a target. The target that is present within the range of sensors but its position is unknown. Observations from Sensor #1 are transmitted to Sensor #A, that fuses it with its own measurements and then retransmits the result to Sensor #B. Sensor #B fuses these information with measurements transmitted from Sensor #2. The final result of the fusion (and estimated target position) is then transmitted to the local user or further retransmitted to the remote user.*

All types of mentioned sensor network configurations have advantages, but also drawbacks. Even though self-organizing distributed networks are more flexible than centralized SN when it comes to adding new sensors and robustness to sensor failures, there appear several problems regarding the sensor fusion and the sensor management. This is a result of none of the sensors having a full information about the information possessed by the other network nodes. On the other hand, centralized networks are easier to manage and allow for easier data fusion.

This is because usually the central node receives information from all the nodes within the SN, so it has a full overview of the scenario. In that case data fusion is much easier and more accurate. The decentralized SN, can be considered as a hybrid of these two types inheriting both the advantages and the drawbacks.

Even though sensor networks might provide large amounts of data, in order to provide consistency in inferred information, the SN has to be precisely calibrated. It is necessary to perform correct detection, classification and target tracking within the area under surveillance. Before stating a proper definition of the calibration problem in Chapter 3, the next two sections will present a generic sensor model equations followed by a presentation of different types of sensors relevant to GSN applications.

2.2 Sensor Model

In general a sensor, gathering a measurement y_k of a target state x_k at time step k, can be defined by the following (known) measurement function

$$y_k = h_k(x_k, \theta_k, \nu_k), \tag{2.1}$$

where y_k might be a scalar or a vector and $h_k(\cdot)$ is the corresponding, usually nonlinear, measurement function at time k. Further, the set of unknown sensor parameters, that will also be called biases in the aspect of calibration, is denoted θ_k. Finally, ν_k represents a noise term distributed according to some distribution $p_k^{\nu}(\cdot)$, that might change over time.

In the problems presented in this thesis a simpler sensor model with additive noise term is considered, where the parameter vector θ is constant over time. The equation describing the model is given by

$$y_k = h_k(x_k, \theta) + \nu_k, \tag{2.2}$$

where the usually nonlinear measurement function $h_k(\cdot)$ depends only on the state of the target x_k and the parameter vector θ. The noise term is additive, distributed according to some known distribution $p_{\nu}(\cdot)$ constant over time k.

In a practical applications it is very often reasonable to linearize the measurement function, in order to simplify its computation. Since bias parameters θ are usually close to zero, and estimates of the state x_k are close to the true values, one can consider a first order Taylor expansion of $h_k(\cdot)$. A model presented in (2.2), linearized around $\hat{x}_k$ and $\hat{\theta}$, takes the form

$$y_k = h_k(\hat{x}_k, \hat{\theta}) + H_k^x(x_k - \hat{x}_k) + H_k^{\theta}(\theta - \hat{\theta}) + \nu_k, \tag{2.3}$$

where

$$H_k^{\theta} = \left.\frac{\partial h_k(x_k, \theta)}{\partial \theta}\right|_{\substack{\theta = \hat{\theta} \\ x_k = \hat{x}_k}}, \qquad H_k^{x} = \left.\frac{\partial h_k(x_k, \theta)}{\partial x_k}\right|_{\substack{\theta = \hat{\theta} \\ x_k = \hat{x}_k}} \tag{2.4}$$

are Jacobians computed for $h_k(\cdot)$ at each time step k, using values of $\theta = \hat{\theta}$ and $x_k = \hat{x}_k$.

2.3 Types of Sensors

Default sensor model presented in (2.2) can be further extended in order to describe different types of measurements collected by the sensors. This section introduces most popular types present in the GSN applications.

There is a large variety of sensor models used to collect measurements within GSN. One can distinguish between sensors providing range to the target, angle to the target, received signal power or just simply a binary signal indicating presence of the target. The following subsections provide a basic description of the measurement function $h(\cdot)$ for the different types of sensors. All models will be considered static functions $h(\cdot)$, with an additive noise term, according to (2.2). A typical set of corresponding bias parameters θ will also be introduced.

It is important to note that this section provides only a very generalized mathematical representations, without going into details behind the signal processing part etc. Further and more detailed discussion over different measurement models (with extensive examples) can be found in Gustafsson (2012).

2.3.1 Linear Range (TOA, Time of Arrival)

In time of arrival sensors, the time difference τ of the time when signal is emitted and received is measured. Knowing the average speed of the signal v in the propagation media (speed of light/sound in air/water), the distance to the observed target x_k can be determined. If both times are known (emission and reception), the model representing linear range to the target can be determined as

$$h^{\text{TOA}}(x_k, \theta) = r_k + \theta^r, \tag{2.5}$$

where

$$r_k = \|p_k - (s + \theta^s)\| \tag{2.6}$$

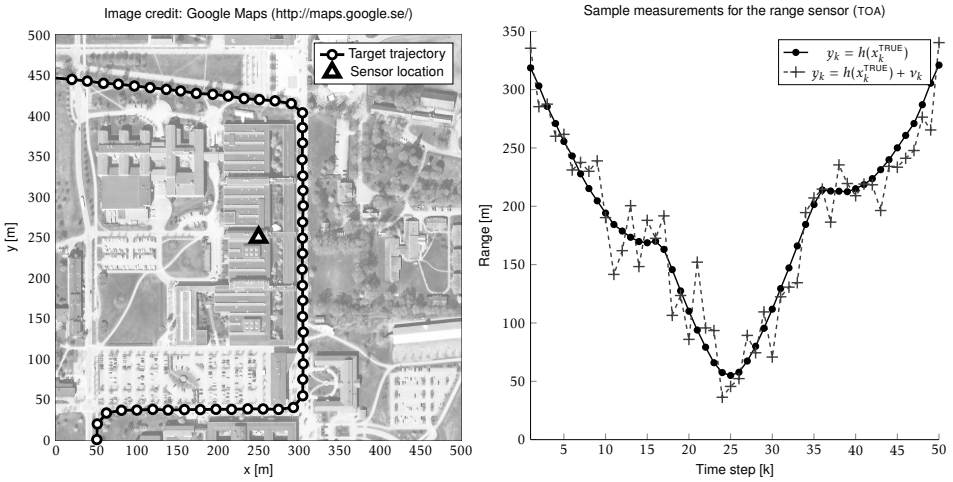
is the range to the observed target, p_k is the position of the target at time k (part of state vector x_k), s is the sensor position, θ^s and θ^r are position and additive range measurement uncertainty biases.

In case the time of emission of the signal is unknown, there appears one more term in the model that needs to be estimated together with the target state vector x_k – the range r^e factor. It corresponds to the emission time multiplied by the known speed of of the signal v. The model takes the form

$$h^{\text{TOA}}(x_k, \theta) = r_k + r^e + \theta^r, \tag{2.7}$$

where as previously $r_k = \|p_k - (s + \theta^s)\|$ and rest of the parameters remain unchanged.

As an example of time or arrival (TOA) sensor a radar or an ultrasound sensor can be considered, where the emitted electromagnetic/acoustic signal is reflected and received by the sensor. Therefore we can determine distance to the target. Sample measurements for TOA sensor are presented in Figure 2.3.



(a) *Sample scenario.* ***(b)*** *Sample measurements.*

Figure 2.3: *Sample one target scenario and corresponding noiseless/noisy measurements for the linear range (TOA) sensor.*

2.3.2 Linear Range Difference (TDOA, Time Difference of Arrival)

In case we do not know when the signal was emitted, in order to remove the unknown emission time r^e, a time difference can be computed (related to the range through known speed of the signal) between target x_k and two sensors. That approach is known as a time difference of arrival (TDOA), and the model is given by

$$h^{\text{TDOA}}(x_k, \theta) = \left(r_k^1 - r_k^2\right) + \theta^d, \tag{2.8}$$

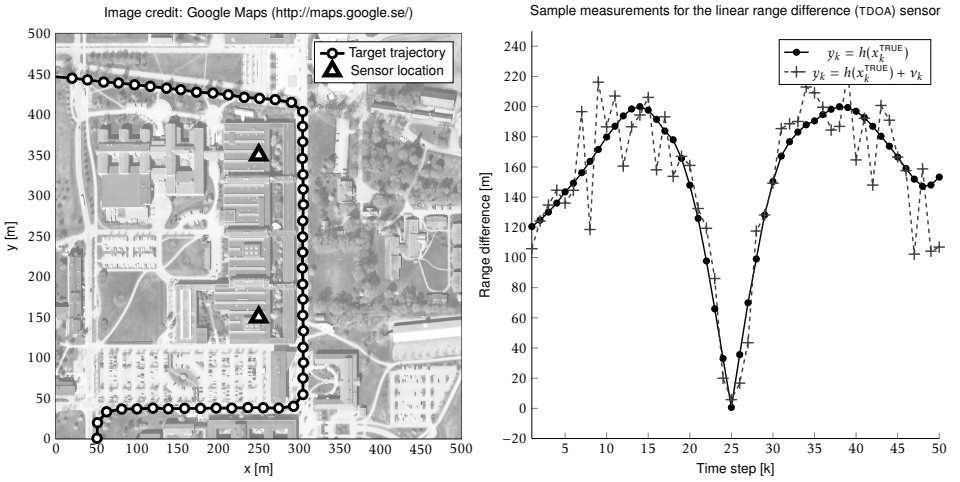
where

$$r_k^1 = \|p_k - (s^1 + \theta^{s^1})\|, \tag{2.9a}$$

$$r_k^2 = \|p_k - (s^2 + \theta^{s^2})\| \tag{2.9b}$$

are distances to the target respectively from Sensor 1 and 2 , s^n is the position of n-th sensor, θ^{s^n} and θ^r are position (of the n-th sensor) and additive range measurement uncertainty biases.

An example of TDOA measurements is a group of acoustic sensors (microphones) detecting sound emitted by the target. Application of signal processing allows to compute time differences in signal reception time between the sensors, so the range difference can be computed. Sample measurements for TDOA sensor are presented in Figure 2.4.



(a) Sample scenario. *(b) Sample measurements.*

Figure 2.4: *Sample one target scenario and corresponding noiseless/noisy measurements for the linear range difference (TDOA) sensor.*

2.3.3 Bearing Measurements (DOA, Direction of Arrival)

The sensor provides bearing (direction, angle) to the target x_k. A simple model, representing direction measurements, is given by

$$h^{\text{DOA}}(x_k, \theta) = \varphi_k + \theta^{\varphi}, \tag{2.10}$$

where

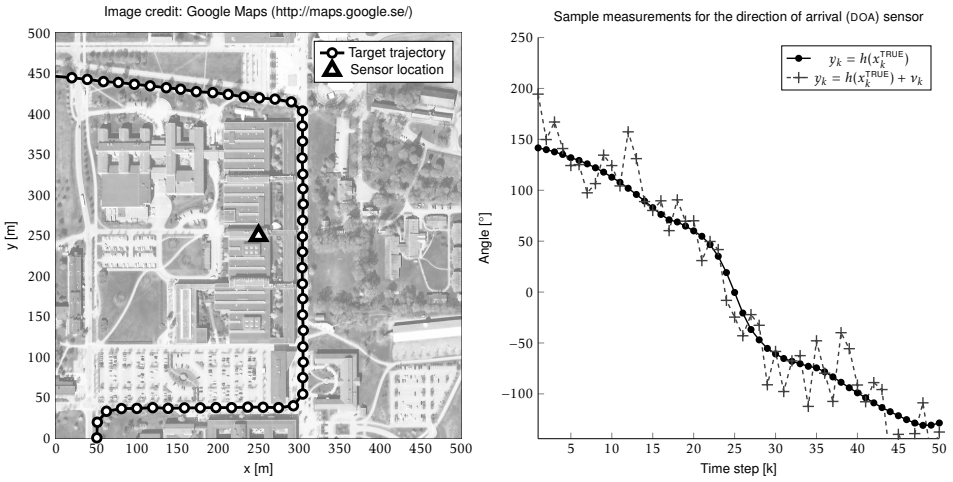
$$\varphi = \arctan\left(\frac{p_k^y - (s^y + \theta^{s^y})}{p_k^x - (s^x + \theta^{s^x})}\right) \tag{2.11}$$

is the angle to the target at time k, p_k^x and p_k^y are x- and y- positions of the target at time k, s^x and s^y are x- and y- positions of the sensor. The parameters θ^{s^x} and θ^{s^y} are the position biases corresponding to the sensor positions respectively and θ^{φ} is a north alignment bias – an uncertainty in the sensor orientation in reference to the global coordinate system.

A typical sensor used to obtain this type of measurements in GSN is a microphone array, where three (or more) microphones are combined together in a geometric shape. Using the differences between the signals and applying signal processing, a bearing to the target can be extracted. Sample measurements for DOA sensor are presented in Figure 2.5.

2.3.4 Radar

A radar sensor is a very popular sensor, especially in application to air surveillance/traffic control, but it is also found in GSN. Radar sensors are usually very



(a) Sample scenario. *(b) Sample measurements.*

Figure 2.5: *Sample one target scenario and corresponding noiseless/noisy measurements for the direction of arrival (DOA) sensor.*

accurate and provide very good measurements. A basic, two dimensional radar sensor combines both range and bearing measurements. It can of course be extended by an elevation in a three dimensional radar case, or by a radial target velocity for a doppler radar.

The main concept underlying the radar is the emission and reception of electromagnetic pulse in a known direction (in a spectrum of microwaves; in case of light spectrum we can distinguish a laser radar – LIDAR). Thus, after applying signal processing to the signal, we can determine bearing and range to the sensor.

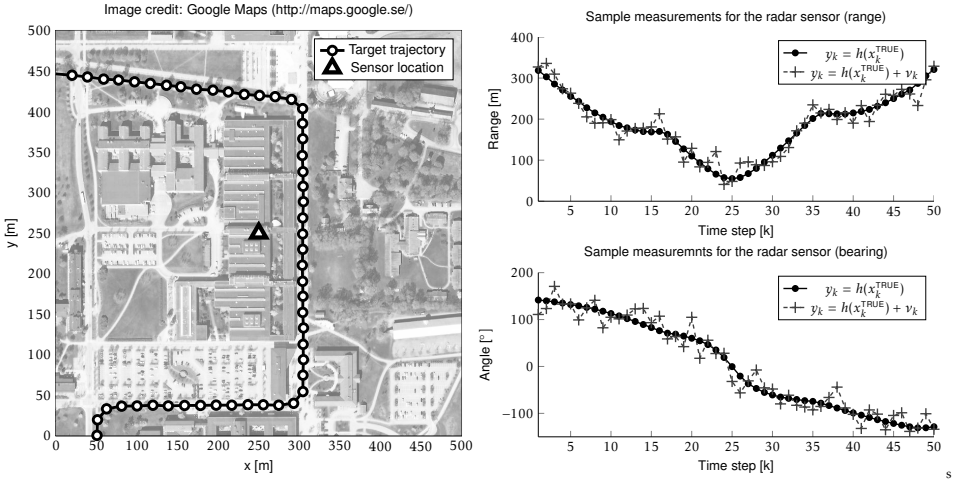
A simple radar sensor model, stated in a very general way, can be defined as

$$h^{\mathrm{RADAR}}(x_k, \theta) = \left[\begin{array}{cc} h^{\mathrm{TOA}}(x_k, \theta) & h^{\mathrm{DOA}}(x_k, \theta) \end{array} \right]^T, \tag{2.12}$$

where according to previous definitions, $h^{\mathrm{TOA}}(x_k, \theta)$ corresponds to the range measurement and $h^{\mathrm{DOA}}(x_k, \theta)$ represents the bearing.

The radar sensor model presented here is relatively simple, and consists only of positioning, range and angle measurement offset biases. Real application situations are more complicated. In brief, the number of bias parameters for one radar sensor can exceed 10 different components including, except position and offset, also biases in antenna pitch and roll or time delays. The typical set of bias parameters for a single three dimensional radar sensor, used in a real application (thanks to the courtesy of Dr Egils Sviestins, Saab AB) can include: *measurement time offset*, *target altitude bias*, *sensor positioning bias*, *sensor axis tilt bias*, *sensor north alignment bias*, *range offset* and *gain bias* together with *antenna pitch and roll bias*.

Sample measurements for radar sensor (range and the corresponding bearing) are presented in Figure 2.6.



(a) Sample scenario. *(b) Sample measurements.*

Figure 2.6: *Sample one target scenario and corresponding noiseless/noisy measurements for the radar sensor.*

2.3.5 Logarithmic Range (Power Measurements, RSS, Received Signal Strength)

The distance between the target and the sensor can also be determined by measuring the received power of a signal emitted by the target. The power usually decays inversely with the distance between the emitter and the receiver. This property can then be exploited to determine the range.

RSS (received signal strength) measurements are obtained by integrating the observed signal (within a certain frequency band) over the specified time window. In order to extract the required frequency the signal can be filtered using a band-pass filter. As a source an acoustic (recorded sound etc.) or an electromagnetic (wifi/3G etc.) signal can be considered.

The sensor model is defined as a logarithmic model

$$h^{\text{RSS}}(x_k, \theta) = (1 - \theta^{\text{G}})P_k^0 - \beta \log(r_k), \tag{2.13}$$

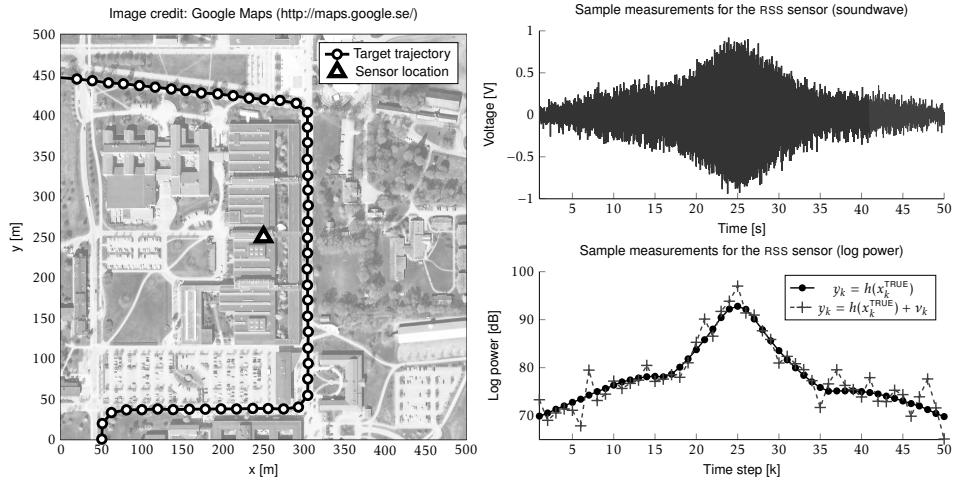
where

$$r_k = \|p_k - (s + \theta^s)\| \tag{2.14}$$

is the distance to the target, P_k^0 is the power emitted by the source (target), p_k is a position of the target at time k, s is the sensor position, θ^s and θ^G are position and gain biases.

The power P_k^0 emitted by the target is not always known. In that case the emitted power can be considered as an another parameter that needs to be estimated. Another option is to combine measurement from a few sensors, in order to eliminate the unknown emitted power (analogously to the TDOA case).

An example of RSS measurements (raw signal and computed power measurement) is presented in Figure 2.7.



(a) *Sample scenario.* ***(b)*** *Sample measurements.*

Figure 2.7: *Sample one target scenario and corresponding noiseless/noisy measurements for the logarithmic range (RSS) sensor.*

2.3.6 Other Sensor Types

Except the basic concepts behind sensor models defined in previous subsections, there are commonly used in the area of GSN. Among them we can distinguish sensors that indicate presence or direction of the moving target, for example

- geophones (or any other seismic sensors) – measure vibrations in the ground and thus indicate presence of the target within the surveilled area,
- infrared sensors – the change in an infrared image can indicate target moving within the range of the sensor,
- magnetometers – used to detect metallic object by measuring changes in the magnetic field, can also indicate the direction of movement and the orientation of the target, and
- cameras – can be used to visually detect, track and classify observed objects.

2.4 Summary

In this chapter a typical sensor network concept was introduced together with the mathematical description of the sensor model. The subject was extended by a presentation of typical sensors used in the GSN applications. Having introduced the measurement model, the calibration problem can be properly stated in Chapter 3.

3

Calibration

As was mentioned earlier, to have a correct calibration of the SN is crucial in order to achieve the desired performance of the network. After deploying the nodes of the network there might still be some uncertainty in the location and orientation of the sensors, depending on how accurate prior information is available. Another cause of inaccuracies is incorrect calibration during the manufacturing process – for example the antenna gain. Also the sensor aging process and the material wear-off might be critical sources of errors.

This chapter provides an introduction to the calibration problem, stated together with a short introduction to the online and the offline bias estimation methods. Next, results of the incorrect calibration on the performance of the SN are presented with corresponding examples. The end of the chapter is dedicated to the analysis of the way the estimation performance is affected by different bias values.

3.1 Problem

In order to provide a basis for the calibration methods presented in further part of the thesis, a formal definition of the calibration problem has to be stated first. Having a set of measurements $Y_K = \{y_k\}_{k=1}^{K}$, collected during time steps $k = 1 \dots K$, one wants to estimate the set of bias parameters θ. In case the states $X_K = \{x_k\}_{k=1}^{K}$ of a reference targets, corresponding to the measurements, are known, we can directly estimate the bias parameters by using one of the available standard methods. A few fundamental algorithms will be described in further chapters.

In a practical application, we are interested in an estimate of θ, given by its mean

value μ_θ and covariance P_θ that represents the uncertainty about our estimate. One might also be interested in finding a posterior distribution $p(\theta|X_K, Y_K)$ of the bias vector θ, given the set of measurements Y_K and corresponding states X_K. In that case, in order to obtain a point estimate, a value maximizing the posterior can be found as

$$\hat{\theta} = \arg\max_{\theta} p(\theta|X_K, Y_K). \tag{3.1}$$

In case there is no target reference available, one also needs to estimate the set of states X_K. Thus, the joint posterior $p(X_K, \theta|Y_K)$ of bias parameters and states is in the scope of interest. Given the distribution, in order to obtain a posterior for biases only, one needs to integrate out the state X_K

$$p(\theta|Y_K) = \int p(X_K, \theta|Y_K) dX_K. \tag{3.2}$$

Then the point estimate can again be obtained by finding the maximum of the posterior

$$\hat{\theta} = \arg\max_{\theta} p(\theta|Y_K). \tag{3.3}$$

Another way to find a point estimate is to use minimum mean square error (MMSE) estimate, that is defined by

$$\hat{\theta}^{\text{MMSE}} = \int \theta p(\theta|Y_K) d\theta. \tag{3.4}$$

3.2 Approaches to Calibration

There are different practical approaches to calibration, depending on what kind of information is available and whether or not the calibration process should be performed online or offline. Online calibration runs in parallel with the operation of the tracking system, whereas in the offline approach the calibration algorithm is run on the complete data set collected during a certain period of time.

In order to properly calibrate a network, a set of raw measurements (collected by the nodes we want to calibrate) is processed together with the state of the targets observed by the network. As targets we can consider beacons with known locations (obtained by GPS or any other accurate method) or a target following a known trajectory. As an example one might consider a car with a mounted GPS receiver, where collected positions, together with the set of raw sensor measurements, are used in the calibration process. The calibration in this case is presented in Figure 3.1.

Unfortunately, reference sources can not be used if their positions are unknown. In that case we need to use targets of opportunity. By that term we consider objects with unknown true state, that has to be estimated during the state estimation process. Estimation, based on the sensor measurements, is thus performed simultaneously with the calibration process. As an example of opportunistic tar-

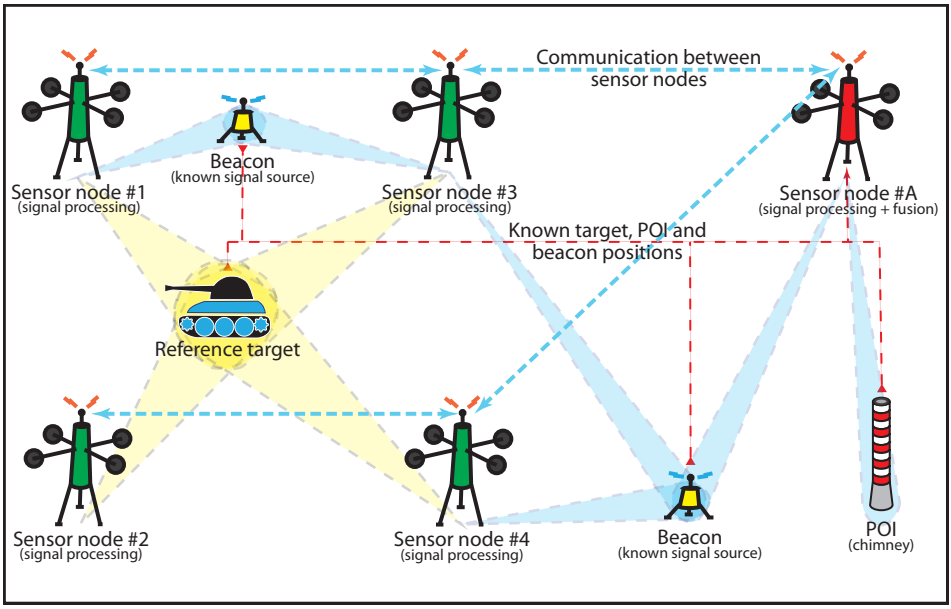


Figure 3.1: *Calibration using known reference targets. Sensors #1 to #4 observe the reference target, (static) calibration beacons and points of interest (POI). Measurements are transmitted to a fusion node, together with the known positions of the reference targets. All the information are then combined together (also with observations collected by Sensor #A) to perform the calibration of the network.*

gets one can consider vehicles, various signal sources (e.g. antennas) and different points of interest (POI) like buildings, trees, chimneys etc. That type of situation is presented in Figure 3.2.

Both calibration approaches (using known and unknown reference targets) can be performed in an online and in an offline manner. Using the online approach, the calibration algorithm usually works in parallel with the tracker to provide bias estimates. They are then used to correct the measurements immediately while the system is running. The offline calibration methods utilizes the complete data set, typically in an iterative manner, to make use of all the information available in the data and provide as accurate bias estimates as possible. Calculated bias values can then be applied to the sensor network to compensate the errors.

3.3 Consequences of Miscalibration

Any uncertainty in sensor position or measurements might cause serious consequences, including degraded accuracy, appearance of ghost tracks and wrong data to track association.

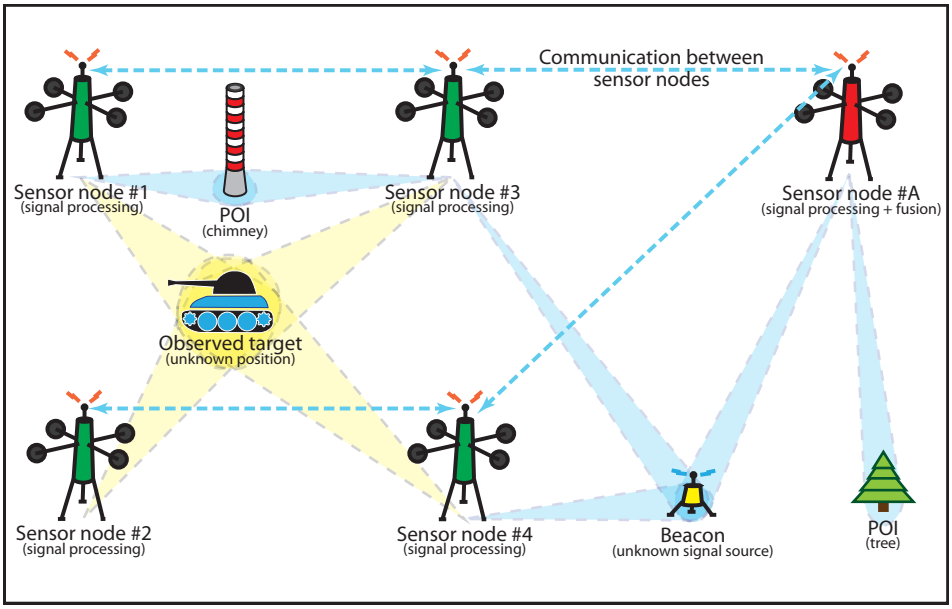


Figure 3.2: *Calibration using targets of opportunity. Sensors #1 to #4 observe the target, (static) beacons and points of interest (POI). All the observed targets are of unknown locations. Measurements are transmitted to a fusion node – Sensor #A, where they are combined together (also with observations collected by Sensor #A). Results of the data fusion – estimated positions of these unknown targets, are then used in the calibration process.*

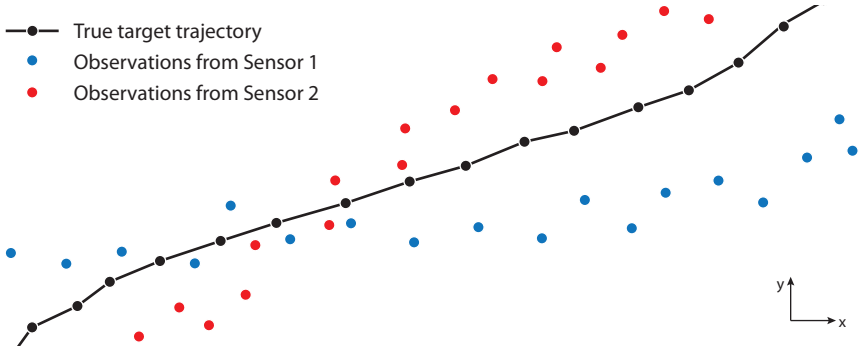


Figure 3.3: *Degraded performance. Due to the large values of measurements biases, the corresponding observations of one target from two sensors are inconsistent, causing the degradation of the overall performance.*

Figure 3.3 presents a sample set of measurements collected by two different sensors observing the same target. Inconsistency in measurements, shown as a large difference between measurements from two different sensors, is a direct result of improper sensor calibration. This situation usually results in a degradation of the tracking process performance and significant increase in the estimation uncertainty.

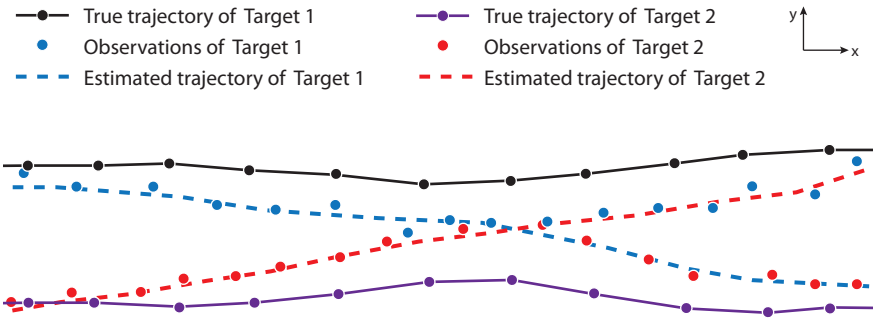


Figure 3.4: *Wrong data-to-track association. Due to the large measurement biases, observations corresponding to one of the targets are incorrectly assigned to the estimated trajectory of the other target.*

Another consequence of incorrect calibration is presented in Figure 3.4. In a case when more than one target is being tracked, measurements are incorrectly assigned to a track they do not belong to. A shift in measured position of the target one, makes it more probable to be wrongly assigned to the track of target two. This might of course cause unexpected problems and errors in the tracking process for both targets, including merging the two targets into one.

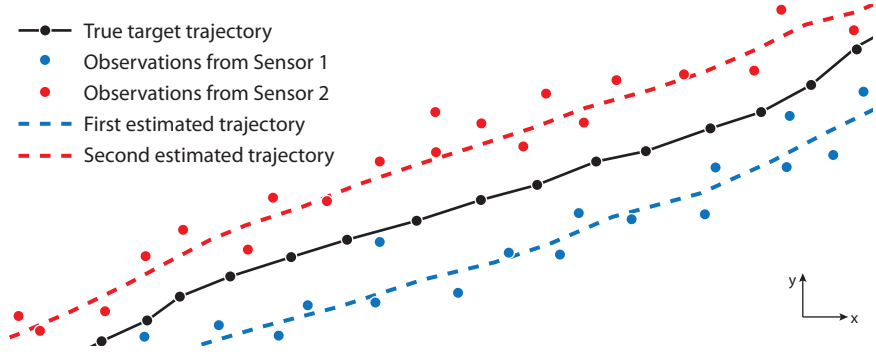


Figure 3.5: *Appearance of ghost tracks. For one true target there appear two separate tracks due to the large measurement biases.*

Finally, so called ghost tracks might appear when all measurements collected by sensors are shifted in the measurement space. This can directly lead to the

appearance of two or more tracks corresponding to only one target. The situation is presented in Figure 3.5.

3.4 Bias Sensitivity Problem

In order to get some understanding of how crucial the problem of correct calibration is, this section will investigate the problem of bias sensitivity. In order to check how the sensor bias (an input) affects the filtering results (an output), a simple scenario with one target and $N = 8$ TOA sensors will be considered.

Measurements are modeled using (2.7) and are assumed to be independent and to have an additive zero mean Gaussian noise with standard deviation $\sigma = 10$ meters.

For each sensor both the position and the north alignment will be biased, and the filtering root mean square (RMS) error of the state estimation will be computed. Biases will be generated according to a zero mean Gaussian distribution with a standard deviation of $\sigma_{xy} = \{0, 5, 10, \ldots, 100\}$ meter for the sensor location bias and $\sigma_r = \{0, 5, 10, \ldots, 100\}$ meter for the range measurement offset bias. To be able to compare the results in a reasonable manner, $N_{\mathrm{MC}} = 50$ Monte Carlo simulations were performed for the each possible pair of the bias standard deviations.

The scenario with an example target trajectory, together with the error visualization for all bias standard deviation combinations is presented in Figure 3.6. As one could expect, the larger the biases are, the more the estimation quality is affected. It can be thus easily concluded, that the process of calibration is crucial for a good performance of the sensor network.

3.5 Summary

In the chapter the problem of calibration was stated in detail. Different approaches to calibration using reference targets/targets of opportunity were presented, both for online and offline application. The effects of improper calibration, including ghost tracks and problems with track to data association were described. The end of the chapter was dedicated to sensitivity analysis, where it was shown how the different values of bias parameters affect the performance of the SN.

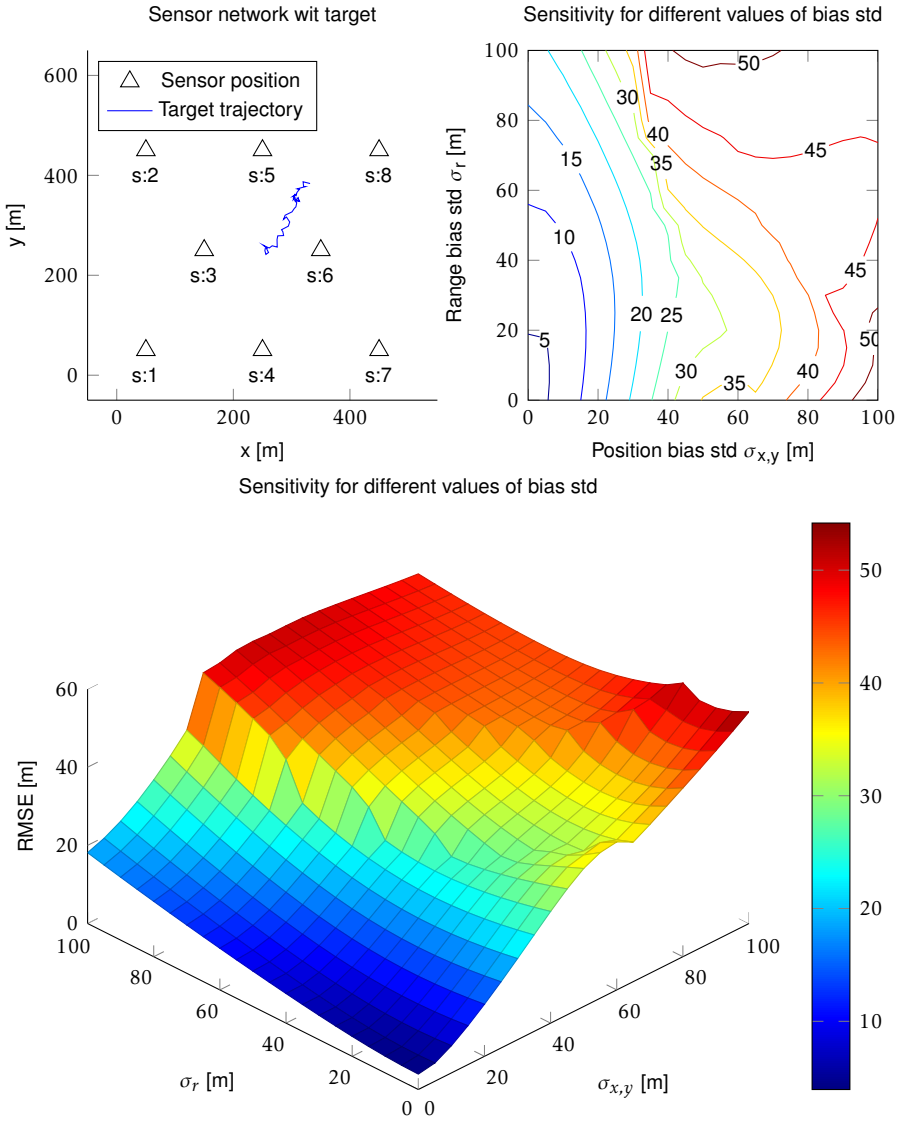


Figure 3.6: *Target state estimation RMS error for different standard deviations of randomly generated bias parameters. Results were generated using 50 Monte Carlo simulations for each set of parameters, with target trajectory randomly generated within the network.*

4

Basic Estimation and Analysis

It was mentioned in previous chapter that there are many approaches to the calibration (online/offline and using known/unknown targets). A correct calibration process implicitly requires knowing of the state of the target (or the calibration beacon) used for the estimation purpose. Thus aside from the methods for the bias parameter estimation, one also needs some techniques to find the states of the targets in case they are unknown.

This chapter briefly introduces basic concepts for both the parameter and the state estimation. Basic least square methods will be initially presented; to be further extended by the regularization and its relation to the Bayesian methods. Subsequent sections will describe basic ideas behind a filtering and smoothing processes, that will provide a necessary background to more sophisticated methods presented in Chapter 5. End of the chapter is dedicated to the Cramér-Rao lower bound (CRLB) together with the parameter observability problem.

4.1 Basic Parameter Estimation Methods

Having defined the problem and the necessary measurement models in Chapter 2, some basic techniques for estimating the bias vector θ can be now introduced. This section will present approaches to solve the problem of calibration in the case when the reference target state is known. The following subsections will describe in detail the nonlinear least squares and regularization concepts (ℓ_1 and ℓ_2, for ill-posed problems) together with its reference to the maximum likelihood and the maximum a posteriori methods.

4.1.1 Least Squares (LS) and Weighted Least Squares (WLS)

One of the basic approaches to the parameter estimation is to use the nonlinear least squares (NLS). The idea underlying the NLS is to minimize the squared sum of residuals $\epsilon_k(\theta)$, where $\epsilon_k(\theta) = y_k - h_k(x_k, \theta)$, w.r.t. the unknown parameter θ, in order to obtain its estimate. In case the measurement noise covariance $R_k = cov(\nu_k)$ is known, it can be used as a weighting factor, and the method takes a form of the nonlinear weighted least squares (NWLS) (Gustafsson, 2012). The estimate is obtained by finding the minimizing argument of

$$\hat{\theta}^{\mathrm{NWLS}} = \arg\min_{\theta} V^{\mathrm{NWLS}}(\theta), \tag{4.1}$$

where $V^{\mathrm{NWLS}}(\theta)$ is a cost function, representing a weighted squared sum of the residuals, and is defined as

$$V^{\mathrm{NWLS}}(\theta) = \frac{1}{2}\sum_{k=1}^{K} \epsilon_k^T R_k^{-1} \epsilon_k = \frac{1}{2}\sum_{k=1}^{K} (y_k - h_k(x_k, \theta))^T R_k^{-1} (y_k - h_k(x_k, \theta)) \tag{4.2}$$

In case there is no weighting ($R_k = I$), the problem simplifies to the NLS problem.

In a linear case when the function $h_k(\cdot)$ is linear (i.e. $h_k(\cdot) = H_k x_k$), the method takes a form of the least squares (or weighted least squares, WLS, for known R_k). Both the regular and the weighted estimates are then unbiased. WLS is also the best linear unbiased estimator (BLUE), which means that the covariance of estimates must follow $cov(\hat{\theta}^{\mathrm{WLS}}) \leq cov(\hat{\theta}^{\mathrm{LS}})$ (Gustafsson, 2012).

4.1.2 Regularization (ℓ_1 and ℓ_2)

In a situation when the NWLS problem is ill-posed (there appear problems with observability etc.), a typical approach to make it well-posed is to apply the regularization. In the calibration framework it relates to choosing those bias vector elements, that are well determined by the data and thus should be estimated.

Another motivation to use the regularization is to control the overfitting of the model to the data and/or to imply sparsity. Especially when the measurement data are not informative enough, it is sometimes reasonable to search for as simple model as possible. The regularization term (added in the form of a penalty to the cost function) allows us to control the solution complexity and the number of elements from the θ vector, we wish to estimate. This is done by introducing λ – a so called regularization parameter, that controls the penalty impact on the estimation. In this section two most popular types of regularization will be considered – the so called ℓ_1 and the ℓ_2 regularization.

The cost function, defined in (4.2), with an extra penalty term corresponding to the $\ell_{j=\{1,2\}}$ regularization, takes the form of

$$V^{\ell_j}(\theta) = V^{\mathrm{NWLS}}(\theta) + J_j(\theta), \tag{4.3}$$

where $J_j(\theta)$ is a penalty term, that for ℓ_2 regularization takes the form of

$$J_2(\theta) = \frac{1}{2}\lambda \|\theta\|_2, \quad \text{where} \quad \|\theta\|_2 = \theta^T\theta = \sum_{n=1}^{N} \theta_n^2, \tag{4.4}$$

where θ_n is the n-th element of a bias vector. For ℓ_1 regularization, the penalty term is defined as

$$J_1(\theta) = \frac{1}{2}\lambda \|\theta\|_1, \quad \text{where} \quad \|\theta\|_1 = \sum_{n=1}^{N} |\theta_n|. \tag{4.5}$$

The main difference between these two regularization methods comes from the shape of regularization term. In the case of ℓ_2, the penalty forces some of the estimated elements of θ to take values close to zero (depending what value of λ was chosen). On the other hand, for the ℓ_1 regularization, some of these bias vector elements take values exactly equal to zero – the penalty implies sparsity. This behavior is called *shrinkage*, and is used to shrink some of the parameters towards zero. Further detailed description of that property can be found for example in Bishop (2006) or Hastie et al. (2005). An elaborate general discussion on regularization methods for system identification and signal processing problems can be found in thesis by Ohlsson (2010) or in the book by Hastie et al. (2005).

As we will see later, inclusion of the regularization terms corresponds to the application of a (non-uniform) prior to the maximum likelihood estimation problem.

4.1.3 Maximum Likelihood (ML)

In a case the measurement noise v_k is Gaussian with a known covariance matrix R_k, the NWLS estimate corresponds to the maximum likelihood estimate. The ML estimate is defined as

$$\hat{\theta}^{\mathrm{ML}} = \arg\max_{\theta} p(Y_K|\theta). \tag{4.6}$$

Using the fact that a logarithm is an increasing function, the above is equivalent to minimizing the negative log of $p(Y_K|\theta)$, thus

$$\hat{\theta}^{\mathrm{ML}} = \arg\min_{\theta} -\log p(Y_K|\theta). \tag{4.7}$$

The negative log likelihood ($-\log p(Y_K|\theta)$), is directly related to the NWLS cost function through

$$-\log p(Y_K|\theta) = C + V^{\mathrm{NWLS}}(\theta), \tag{4.8}$$

where C is some constant term (Gustafsson, 2012).

4.1.4 Maximum a Posteriori (MAP)

In case one has some prior information about the unknown parameter vector, it is reasonable to use this knowledge to obtain better estimates. Thus, introducing the distribution $p(\theta)$, that describes the prior knowledge about the θ, one can obtain maximum a posteriori estimate (MAP). In that case, instead of (maximizing)

the likelihood one maximizes the posterior distribution. The MAP estimate of the parameter vector, is defined by

$$\hat{\theta}^{\text{MAP}} = \arg\max_{\theta} p(\theta|Y_K) = \arg\min_{\theta} -\log p(\theta|Y_K). \tag{4.9}$$

Using the Bayes' theorem to decompose the posterior, we get

$$p(\theta|Y_K) = \frac{p(Y_K|\theta)p(\theta)}{p(Y_K)} = \frac{1}{\tilde{C}} p(Y_K|\theta)p(\theta), \tag{4.10}$$

where $\tilde{C}$ is a normalization constant independent of θ. We can identify the negative log of the posterior for θ as

$$-\log p(\theta|Y_K) = \tilde{C} - \log p(Y_K|\theta) - \log p(\theta), \tag{4.11}$$

which is the sum of the negative log likelihood (as in the ML case) and the negative log of the prior.

Choosing the right prior distribution is crucial for the practical estimation process. Usually one selects its parameters according to some knowledge available, for example accuracy of a sensor positioning or a gain tuning. It can easily be shown that choosing certain prior is directly connected to the ℓ_1 and ℓ_2 regularization. For example using a Laplace distribution corresponds to ℓ_1, and choosing a Gaussian distribution corresponds to ℓ_2 regularization. For further details on the Gaussian and the Laplace distributions see the Appendix A.

Assuming bias parameters θ_n are independent, one can define the priors as a product of independent distributions according to

$$\text{for } \ell_2 : p(\theta) = \prod_{n=1}^{N} \mathcal{N}(\theta_n|\mu_n, \sigma_n^2), \tag{4.12a}$$

$$\text{for } \ell_1 : p(\theta) = \prod_{n=1}^{N} \mathcal{L}(\theta_n|\mu_n, \beta_n), \tag{4.12b}$$

where μ_n corresponds to the mean and σ_n^2 is a variance of the Gaussian distribution $\mathcal{N}(\cdot)$. For the Laplace distribution $\mathcal{L}(\cdot)$, μ_n is defined as a location parameter and $\beta_n \geq 0$ is a scale parameter. Having defined the priors, and by assuming $\mu_n = 0$, the negative log posterior can be written as

$$-\log p(\theta|Y_K) = \bar{C} + V^{\text{NWLS}}(\theta) + J_j(\theta), \tag{4.13}$$

where $\bar{C}$ is a constant and $V^{\text{NWLS}}(\theta)$ is a NWLS cost function. Regularization (penalty) terms $J_j(\theta)$ can then be identified for both regularization methods, as

$$\text{for } \ell_2 : J_2(\theta) = \frac{1}{2} \sum_{n=1}^{N} \lambda_n \, \theta_n^2, \tag{4.14a}$$

$$\text{for } \ell_1 : J_1(\theta) = \frac{1}{2} \sum_{n=1}^{N} \lambda_n \, |\theta_n|. \tag{4.14b}$$

The regularization parameter for that case is directly related to the variance of Gaussian distribution for ℓ_2, as $\lambda_n = 1/\sigma_n^2$, and to the scale parameter of Laplace distribution for ℓ_1, as $\lambda_n = 1/\beta_n$. In case the regularization parameter takes the same value for each corresponding θ_n, ($\lambda = \lambda_1 = \ldots = \lambda_n$), one gets situation analogous to the one presented in (4.4 - 4.5).

The type of priors that we have considered above, with mean (for the Gaussian) or location parameter (for the Laplace) equal to zero, are the so-called *shrinkage priors*. That kind of priors is used when we would like to imply the sparsity, since they have an ability to shrink some of the estimated parameters towards zero. The situation is analogous to the shrinkage process via regularization, introduced in Section 4.1.2 or presented in Hastie et al. (2005).

When the bias parameters are correlated, we might consider using a multivariate prior distribution. An example might be a Gaussian prior, where the normal distribution is then defined as $p(\theta) = \mathcal{N}(\theta|\mu, \Sigma)$ with μ being the mean and Σ being the covariance matrix for the prior of θ. In that case there is no longer direct correlation between regularization and application of the prior, because the corresponding λ-s cannot be precisely identified. It is due to the fact that Σ is not a diagonal matrix anymore.

There of course in practice exist different prior distributions that can be applied, as well as the regularization methods, but they are out of the scope of this thesis.

4.1.5 Existence of the Closed Form Solution

There exist different approaches to minimization of the cost function for both the regularized and the non-regularized problems. For the linear case (LS and WLS) it is possible to find a closed form solution by taking a gradient and equalling it to zero. Since there usually does not exist a closed form solution to the nonlinear case, the optimization problems can be solved, e.g., using one of the optimization techniques (like the Gauss-Newton) or a dedicated toolbox, e.g. CVX [1] or YALMIP[2]. It is also possible to linearize the measurement function and turn the problem into WLS, using the Taylor series expansion, according to (2.3).

An analogous situation applies to the ML case. In the linear and Gaussian case there exists direct closed form solution, analogous to the WLS solution. Using the idea for a nonlinear case, an approximate (closed form) solution can be obtained using local linearization. In other cases, of course depending on a likelihood form, the problem usually does not have a direct solution, and the numerical optimization methods might be necessary to apply.

For the regularized problem, a closed form solution can be found for linear ℓ_2 case. For ℓ_1, there is usually no direct solution. In that case numerical optimization method needs to be used.

[1] CVX toolbox can be found at: `http://cvxr.com/cvx/`

[2] YALMIP toolbox can be found at: `http://users.isy.liu.se/johanl/yalmip/`

The situation as above is analogous to the MAP problem, where the closed form solution can be obtained only for the linear case with a normal prior distribution.

4.2 State Estimation Techniques

The problem of estimating the latent state of the target generating the observations from multiple sensors is one of the basic problems of data fusion. This section describes two principal approaches to state estimation: filtering and smoothing. Both methods estimate the state based on the gathered measurements. The main difference is that filtering is an online process, where the state estimate $\hat{x}_k$ is obtained using a stream of noisy measurements up to current time k. The smoothing is a batch process, where also the future observations, up to time $k+m$, $m \geq 1$, are used in order to obtain more accurate estimates.

For the purpose of this section, a general nonlinear model for the target (motion model) and observations (measurements model) is assumed, and is defined as

$$x_{k+1} \sim p(x_{k+1}|x_k), \tag{4.15a}$$
$$y_k \sim p(y_k|x_k). \tag{4.15b}$$

The transition process between x_k and x_{k+1} is described by $p(x_{k+1}|x_k)$, under the assumption of first order Markov property on the dynamics. The observation model states, that the measurement y_k at k-th time step depends on the current state x_k.

The full, nonlinear model, that describes both sensors and targets can then be defined by extending the SN model presented in (2.1). After adding the motion model it can be restated as

$$y_k = h_k(x_k, \theta, \nu_k), \tag{4.16a}$$
$$x_{k+1} = f_k(x_k, \eta_k), \tag{4.16b}$$

where y_k is the measurement, x_k is the state of the target at time k, $h_k(\cdot)$ and $f(\cdot)$ are (known) measurement and motion models at time k respectively, θ is a parameter vector and ν_k and η_k are measurement and process noises respectively. Both noises are distributed according to some distribution $p_k^{\nu}(\cdot)$ and $p_k^{\eta}(\cdot)$.

For the purpose of this thesis, a model with the additive Gaussian noise will be considered. The model in (4.16) takes hence a form of

$$y_k = h_k(x_k, \theta) + \nu_k, \tag{4.17a}$$
$$x_{k+1} = f_k(x_k) + \eta_k. \tag{4.17b}$$

For the simplicity of the presentation of filtering and smoothing problems, it is assumed that the set of bias parameters of the SN is known – the network is correctly calibrated.

4.2.1 Filtering

This section describes a Bayesian approach to nonlinear filtering problem. Having a set of measurements $Y_k = \{y_i\}_{i=1}^k$ gathered by all sensors, a task is to estimate the set of corresponding target states $X_k = \{x_i\}_{i=1}^k$. Thus we are interested in finding a sequential approximation of a joint distributions of states given measurements, $p(X_k|Y_k)$ and a likelihood $p(Y_k)$. Having those distributions one can directly sample the entire trajectory X_k, or just obtain a distribution for current state x_k, by applying the marginalization.

In the literature filtering is usually understood as a process of estimating the conditional distribution $p(x_k|Y_k)$ of the current state, based on a set of observations up to the time step k, rather than sampling from $p(X_k|Y_k)$. In practical applications we are usually interested in obtaining:

- The **filtering** distribution: $p(x_k|Y_k)$, where the state at time k is estimated based on measurement up to time k.
- The **prediction** distribution: $p(x_{k+m}|Y_k)$, where the state at future time $k+m$, $m \geq 1$, is predicted, based on the set of measurements Y_k.

In order to obtain those posteriors, a general Bayesian update recursion can be applied, which for the model defined in (4.15) consists of two main steps:

- **Time update**: in this step a prediction x_{k+1} – for the next time instant, is computed using current estimate of x_k. It is done by marginalizing x_k through integration from

$$p(x_{k+1}|Y_k) = \int p(x_{k+1}, x_k|Y_k)dx_k = \int p(x_{k+1}|x_k)p(x_k|Y_k)dx_k, \tag{4.18}$$

  where the Markov property of the model was used.

- **Measurement update**: having the prediction $p(x_{k+1}|Y_k)$ from the previous step, when a new measurement y_{k+1} at time $k+1$ arrives, it is reasonable to incorporate that extra information to get the posterior for x_{k+1} given all the measurements up to the current time. A measurement update, using Bayes' theorem, is given by

$$p(x_{k+1}|Y_{k+1}) = \frac{p(y_{k+1}|x_{k+1})p(x_{k+1}|Y_k)}{p(y_{k+1}|Y_k)}, \tag{4.19}$$

  where

$$p(y_{k+1}|Y_k) = \int p(y_{k+1}|x_{k+1})p(x_{k+1}|Y_k)dx_{k+1}, \tag{4.20}$$

  is considered as a normalization constant independent of x_{k+1}.

- **Initialization**: the filtering recursion is initiated by $p(x_1|y_0) = p(x_0)$.

A solution to the filtering problem is very broadly described in the literature. Among practical applications of Bayesian recursion one can distinguish most popular filtering methods:

- **Kalman Filter** (KF): provides an optimal solution to the filtering problem in case, when both process model and measurement model are linear together with an additive Gaussian noise. Thanks to the linearity there exists an exact, closed form solution to each subsequent step of the filtering process.
- **Extended Kalman Filter** (EKF): An extension to the Kalman filtering problem to nonlinear models. In this case modeling the nonlinearity is addressed through the first or second order Taylor series expansion of a motion model and/or measurement function.
- **Unscented Kalman Filter** (UKF): provides a solution for the nonlinear problem. In this case an unscented transform is used, where the distribution is approximated by a set of deterministic (so called sigma) points. The sigma points are then propagated through the non-linear functions (motion model or measurement function), and then fit with a Gaussian distribution.
- **Point Mass Filter** (PMF): is not limited to any linear or Gaussian model, since it provides a numerical approximation to the nonlinear problem. The underlying idea is to apply a grid over the state space in order to compute the posterior. The main limiting factor is the curse of dimensionality for higher state dimensions.
- **Particle Filter** (PF): provides a similar solution methodology of solving the nonlinear filtering problem as a PMF, but with an adaptive stochastic grid over the state space. In this case grid points are chosen automatically according to their relevance. This method also suffers from the curse of dimensionality, but there are some approaches to particularly solve the problem. One of these approaches is marginalization, where the state is split into a part that enters the model in a linear and a part that enters in a nonlinear way. The PF is then used only to estimate the nonlinear part, and the KF can be used to estimate the linear part (see Andrieu and Doucet (2002) or Nordlund and Gustafsson (2009)).

Detailed description of the methods mentioned above (together with derivation process etc.) can be found in basic literature in this area, including Bar-Shalom et al. (2004) and Gustafsson (2012). An introduction to particle filtering can be found in one of the tutorial papers (Arulampalam et al. (2002), Cappé et al. (2007) or Gustafsson (2010)) or books (Ristic et al. (2004) or Gustafsson (2012)).

In order to retrieve an estimate $\hat{x}_{k|k}$ of the state vector (and its covariance $\hat{P}_{k|k}$) from the posterior $p(x_k|Y_k)$, a minimum mean square error (MMSE) estimate of the state can be used, defined by

$$\hat{x}_{k|k}^{\text{MMSE}} = \int x_k p(x_k|Y_k) dx_k, \tag{4.21}$$

for the mean, and

$$\hat{P}_{k|k}^{\text{MMSE}} = \int (x_k - \hat{x}_{k|k}^{\text{MMSE}})(x_k - \hat{x}_{k|k}^{\text{MMSE}})^T p(x_k|Y_k) dx_k, \tag{4.22}$$

for the covariance of the state vector estimate.

4.2.2 Smoothing

Filtering, as presented in previous section, is a process of estimation of the current state of the target based on set of measurements up to the current time step. The idea underlying the problem of smoothing is similar, but here we are interested in the estimation of a state distribution given also all the measurements up to some time $K \geq k$. The estimates tend to be smoother since more information is used. This section will briefly describe basic concept and methods for obtaining smoothed estimates.

Analogous to the filtering problem there are two possible concepts regarding smoothing. First is to sample from joint posterior $p(X_K|Y_K)$, where X_K is a set of target states estimated based on the complete set of measurements Y_K. Another approach is to compute marginal distribution $p(x_k|Y_K)$ for each k. The marginals can be computed in principle by integrating out the remaining states from the joint posterior.

In order to obtain smoothed estimates for time k, one can either use only few future measurements up to time $k + L \leq K$, or a full set of measurements up to time K. The solution to the first problem is given by:

- **Fixed-lag smoothing**: The idea is to perform smoothing using a fixed-lag L. Thus one is interested in using number of L future measurements in order to compute the posterior density $p(x_{k-L}|Y_k)$ for the estimate of x_{k-L} using measurements up to time k. The estimation process is thus lagged and it is possible to use the future knowledge about measurements.

Among methods, where we use the full set of measurements in a batch manner, we can specify two example solutions:

- **Forward-backward recursion**: Is a two step process proposed in (Doucet et al., 2000). First perform filtering forward and then, having estimated set of states $\{\hat{x}_{k|k}\}_{k=1}^{K}$, perform smoothing backwards in order to obtain the smoothed estimates $\{\hat{x}_{k|K}\}_{k=1}^{K}$. In order to perform forward-backward recursion one has to:
  - Compute and store, during the filtering step, a set of marginal distributions $p(x_k|Y_k)$,
  - Compute backwards $p(x_k|Y_K)$ using the set of marginals computed in previous step. Backward recursion is initiated with the final state distribution $p(x_K|Y_K)$.
- **Two filter formula:** The idea (Bresler, 1986) underlying the two filter formula is an introduction of a set of artificial probability distributions $\tilde{p}_k(x_k)$, together with the the joint distribution $\tilde{p}_k(x_{k:K}|y_{k:K})$, which are related to the backward information filter. The basic idea (omitting the details) is as follows:
  - Use standard forward recursion to obtain a set of marginal distributions $p(x_k|Y_{k-1})$,

- Use a backward recursion to obtain a set of $\tilde{p}(x_k|y_{k:K})$ distributions,
- Combine both above distributions in order to obtain smoothed posterior $p(x_k|Y_K)$.

As for the filtering, there exist different approaches to implementation of smoothing. Each filtering method has its complimentary smoothing method – for Kalman filter one has Rauch-Tung-Striebel equations (so called Kalman smoother), with an extension to the nonlinear problem analogous to the EKF. Smoothing is also solved using particle methods, where the Particle Filter is used in conjunction with a Particle Smoother. Further discussion about smoothing methods can be found for example in Cappé et al. (2007) or Doucet and Johansen (2009).

4.3 Simultaneous State and Parameter Estimation

So far the problems of calibration and state estimation were presented separately. In real applications it is most common, that neither the target trajectory is known nor the sensor network is correctly calibrated. In this thesis a deeper look at that problem will be taken in Chapter 5 and particularly in Part II. So far, in order to introduce the problem, a simple method to estimate both unknown variables is presented – an approach to filtering with an augmented state vector.

In order to estimate both state and bias parameters, a basic idea is to extend the state vector x_k with unknown bias vector θ, to form the augmented vector

$$x_k^{\text{AUG}} = \begin{bmatrix} x_k \\ \theta \end{bmatrix}. \tag{4.23}$$

Then one of the filtering methods can be applied, in order to obtain a posterior for augmented state $p(X_K^{\text{AUG}}|Y_K) = p(X_K, \theta|Y_K)$, where $X_K^{\text{AUG}} = \{x_k^{\text{AUG}}\}_{k=1}^K$.

If the bias parameters change over time, and we have some understanding about the character of these changes, it is wise to extend the bias vector model with

$$\theta_{k+1} = f_\theta(\xi_k), \tag{4.24}$$

where $f_\theta(\xi_k)$ is some linear/nonlinear transition function of the bias vector and ξ_k is a process noise with some known distribution $p_k^\xi(\cdot)$.

When we want to use a particle filter in the estimation process, and biases enter the model in a linear Gaussian way, it is possible to use the marginalization. In that case a PF is used to estimate the state and a KF is used for the linear part. The method has been known as marginalized particle filter (MPF) or Rao-Blackwellized PF. Further details can be found e.g. in Andrieu and Doucet (2002) or in Nordlund and Gustafsson (2009).

Although the approach is simple, the method suffers some practical problems. The augmented vector might be very large in a multiple target and multiple sensor scenario, which might cause estimation problems. When the KF is applied, the necessity of computing (and inverting) large covariance matrices might de-

grade performance drastically. Also lack of the observability can cause numerical problems.

The approach presented above can be for example found in (Dhar, 1993). In the literature there also exist different approaches to decompose the problem of simultaneous estimation of state and parameter vector into two nested subsequent problems, presented for example in Ignagni (1981) or Sviestins (1999). In the further part of the thesis a main focus will be placed on decomposition of the problem into two sub-problems: state estimation and calibration, that can be solved separately but in parallel.

4.4 Basic Analysis

This section describes a problem of parameter observability and estimation performance, taking into account typical applications in the ground sensor networks area. The problem of parameter observability and ways to improve it are presented. Further, the Cramér-Rao lower bound is introduced, that is a measure of a lower achievable bound on the variance of parameter estimates.

4.4.1 Parameter Observability

The problem of observability for both parameters and the states is a serious problem in the area of GSN. The number of sensors observing one particular target at the same time might be quite low (usually due to a limited range of the sensors). Also trajectories of targets of opportunity might not be very informative, for example when targets are far from the sensor or move along only one path. Both of these result in a low amount of information contained in the measurements.

Another problem corresponds to the form of the measurement function, where one type of biases can be replaced with other. For example in the case of time of arrival measurement, a range measurement offset can be easily replaced with the uncertainty of position in the x- and y- direction. The situation is graphically explained in Figure 4.1 The problem of observability can be explained in detail by using the problem of computing a WLS estimate as an example. Consider a measurement function linear with respect to the bias parameter vector

$$y_k = H_k(x_k)\theta + \nu_k. \tag{4.25}$$

It can be reformulated to a compact form

$$\bar{y} = \bar{H}\theta + \bar{\nu}, \tag{4.26}$$

where $\bar{y} = [y_1^T \dots y_K^T]^T$ is a (stacked) vector, containing all measurements for all time instances k, $\bar{H} = [H_1(x_1) \dots H_K(x_K)]^T$ and $\bar{\nu} = [\nu_1^T \dots \nu_K]^T$. A solution to WLS problem, being a solution to maximization of the cost function defined in (4.2), is given by the following estimate

$$\hat{\theta} = \left(\sum_{k=1}^{K} H_k(x_k) R^{-1} H_k^T(x_k) \right)^{-1} \left(\sum_{k=1}^{K} H_k(x_k) R^{-1} y_k \right) = \left(\bar{H}\bar{R}^{-1}\bar{H}^T \right)^{-1} \bar{H}\bar{R}^{-1}\bar{y}. \tag{4.27}$$

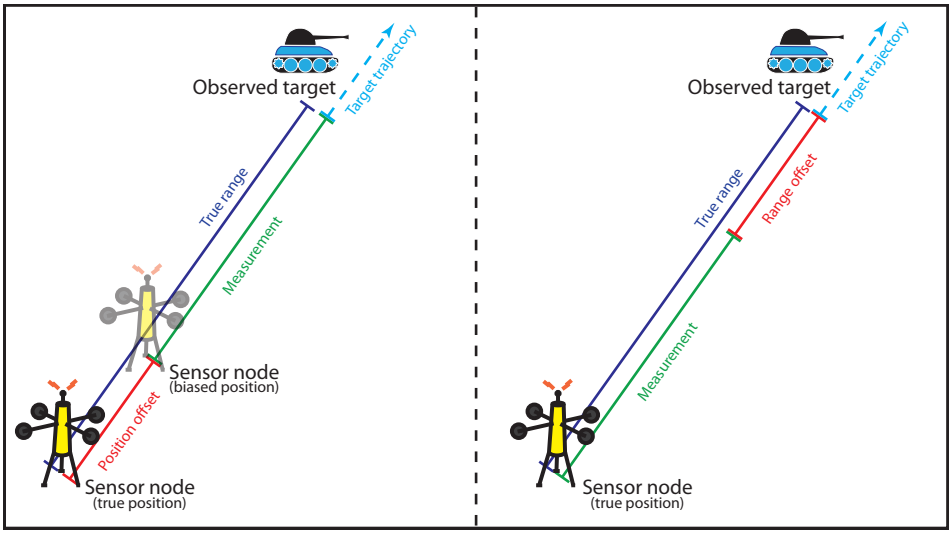


Figure 4.1: *Interchangeability of biases. For the range sensor, in the case of presented target trajectory, it is impossible to uniquely determine whether inaccuracy in measurements comes from the position offset (left side of the figure) or from the range offset (right side of the figure). A change in x- and y-sensor location is then equivalent to the additive measurement bias, making it unobservable.*

In a shorter form

$$\hat{\theta} = B^{-1} A(\bar{y}), \tag{4.28}$$

where we can identify the term $B = \bar{H}\bar{R}^{-1}\bar{H}^T$ and the term $A(\bar{y}) = \bar{H}\bar{R}^{-1}\bar{y}$.

For a unique solution to exist, the problem of maximization of the cost function must be well-conditioned. In other words the B matrix must be invertible. It means that it must have a full rank, so $rank(B) = M$, where M is the size of θ vector. A problem appears when there are too many degrees of freedom for the bias parameters, making a unique solution unavailable.

There are different ways to restrict the freedom of the bias parameters. One of the ideas is to use known road constraints (on the motion model) in order to eliminate some degrees of freedom – otherwise one might encounter rotations and displacements of sensor networks that are unobservable. In case the number of measurements is low, and not sufficiently informative, it might happen that it is impossible to get reasonable estimates. Here one of the ideas is to use regularization, presented earlier in Section 4.1.2, in order to force some of the parameters towards zero and only estimate these ones for which the measurements are informative enough. An equivalent way of the regularization is to apply priors to the parameters in a Bayesian manner.

The main problem, both with the regularization and with the application of the priors is usually the lack of knowledge of the prior (or the corresponding regularization parameters). In the next chapter the problem of finding the parameters of a prior for the bias vector will be described with a possible solution using the evidence method. It is based on the maximization of the probability of the data with respect to the hyper-parameters we want to find.

In the next subsection the Cramér-Rao lower bound (CRLB) will be presented, which is an analytical measure of the lower bound on achievable estimation accuracy for the problem of parameter and state estimation for a certain scenario.

4.4.2 Cramér-Rao Lower Bound (CRLB)

The Cramér-Rao lower bound is a measure of a lower bound on the variance of the estimated parameter, when the estimator is unbiased. It is defined as an inverse of the Fisher information matrix (FIM). The CRLB is to be interpreted as a minimal value of the variance of the unbiased estimator of a parameter we are trying to find. An estimator that achieves the lower bound is the minimum variance unbiased estimator (Gustafsson, 2012), which means that it has achieved a minimum mean squared error. The lower bound for bias parameter θ is thus defined as

$$Cov(\theta) \geq \mathcal{I}^{-1}(\theta), \tag{4.29}$$

where a positive definite Fisher information matrix $\mathcal{I}(\theta)$ is computed using the following formula (Gustafsson, 2012)

$$\mathcal{I}(\theta) = E\left[\left(\frac{d \log p(Y_K|\theta)}{d\theta}\right)\left(\frac{d \log p(Y_K|\theta)}{d\theta}\right)^T\right], \tag{4.30}$$

where $p(Y_K|\theta)$ is a likelihood. Analogously one can define CRLB for state estimates, as

$$Cov(X_K) \geq \mathcal{I}^{-1}(X_K), \tag{4.31}$$

where in this case the CRLB determines the lower bound for the variance of state estimates.

To present an example of how CRLB is distributed along the area covered by a SN, a simple scenario will be considered with $N = 8$ direction of arrival (DOA) sensors. The measurement function considered here is given by (2.10) and measurements are assumed to be independent. Measurement noise here is additive, normally distributed with zero mean and standard deviation taken as $\sigma = 10$ degrees.

Figure 4.2 presents the CRLB for the sensor network example. Upper left plots shows the configuration of the network. Other two plots present two and three dimensional presentation of the minimum achievable RMS for the estimation of the target position (in meters). As was expected, the best achievable performance is within the area covered by the network. The further we drift from the center of the network, the worse the minimum achievable error gets.

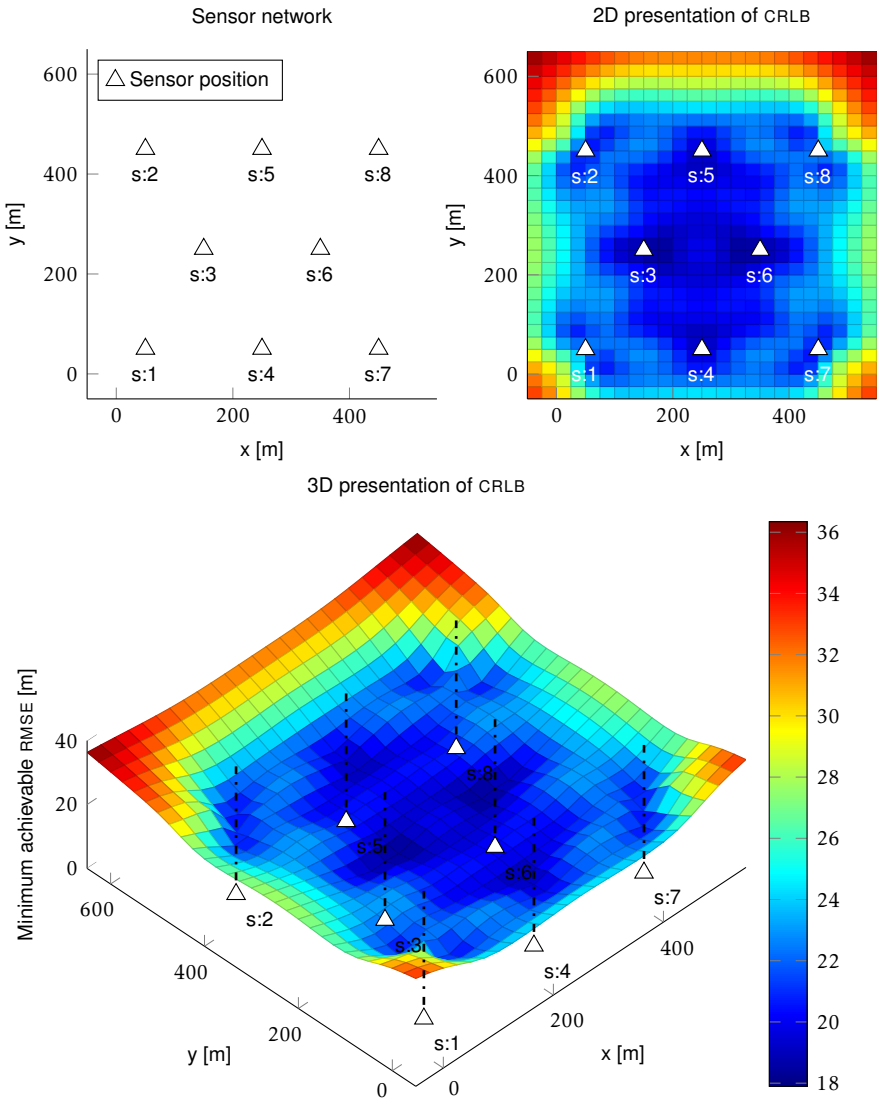


Figure 4.2: *Cramér-Rao lower bound for the position estimates in the DOA sensor network. Upper left plot presents the sensor network configuration. Other two plots present the two and the three dimensional plots of the minimum achievable error for the state estimation performance of the network.*

4.5 Summary

Basic concepts for both the parameter and state estimation were presented in this chapter. In the next chapter the problem of simultaneous estimation of bias parameters and target states, introduced in Section 4.3, will be further extended. It

will be shown that the solution can be found using the expectation maximization (EM) algorithm, since the unknown state can be treated as a latent (unknown) variable.

The next problem to be tackled is a situation of the lack of observability. As was mentioned earlier, one way to solve this is to apply priors to the estimation process. Since the prior knowledge is not always available, next chapter will describe a method of selecting prior parameters based on maximization of evidence function, where the evidence is a measure of how well the proposed model describes the data.

5
Advanced Methods

It was mentioned before that the maximum likelihood methods can be easily applied to the problem of calibration in the situation when the set of target states is known. In a case when both bias parameters and target states are unknown, one can use state vector augmentation, as was presented earlier in Section 4.3. Even though there exist some ways to decouple the problem, in the first part of this chapter an alternative method called expectation maximization (EM) will be presented. It is a two step iterative procedure, used to compute the maximum likelihood (or the maximum a posteriori) estimates in a presence of the unknown (latent) variables.

Another problem that appears very often, in reference to the maximum a posteriori estimate, is the choice of parameters for the prior distribution. Because the prior is driven by a set of hyper-parameters, that are usually unknown, there is a need to estimate them using for example the observation data. Since the case is not trivial, second part of the chapter will be devoted to the evidence approximation (EA) method, that provides a solution to this problem.

Evidence approximation is an iterative method used to simultaneously estimate the parameters and the unknown hyper-parameters. One of the advantages of the method is a fact, that it uses the Occam's razor principle. The razor helps to find an optimal balance between the model complexity and the fit to the measurement data. Using the so-called shrinkage priors also implies the sparsity of a parameter vector.

End of the chapter will be dedicated to a brief introduction to Paper C, where both methods are combined together. A result is a robust, lopped EM-like algorithm for simultaneous state estimation and sparse bias estimation.

5.1 Expectation Maximization (EM) Algorithm

The expectation maximization algorithm (Dempster et al. (1977); McLachlan et al. (2004)) is a technique used to compute the maximum likelihood estimate in a presence of unknown (latent) variables. These variables are observed indirectly, through some function $h : X \rightarrow Y$, in a presence of the observation (measurement) noise. The likelihood for the set of measurements Y_K is governed by the set of parameters θ, which value we want to find. The likelihood for Y_K can be written as

$$p(Y_K|\theta) = \int p(X_K, Y_K|\theta) dX_K, \tag{5.1}$$

where the joint likelihood for measurements and latent variables is given by $p(X_K, Y_K|\theta)$. For the problem of calibration, Y_K represents the set of observations of the target state X_K. Observations are collected using a measurement function $h(x_k, \theta)$, in a presence of the measurement noise. Set of bias parameters θ is then the unknown we want to estimate.

In the EM algorithm an important assumption is, that the maximization of the joint likelihood $p(X_K, Y_K|\theta)$ should be much simpler than the direct maximization of the incomplete data likelihood $p(Y_K|\theta)$. It might be, for example, that even though the joint likelihood belongs to the exponential family, the marginal $p(Y_K|\theta)$ typically does not (Bishop, 2006).

The method comprises of two subsequent step: expectation (E) step, where the expectation of the joint log likelihood is computed using some initial value of the bias vector $\hat{\theta}^{old}$. The resulting expectation is expressed as $Q(\theta, \hat{\theta}^{old})$, which is a function of the unknown bias vector. It is then maximized in maximization (M) step in order to obtain a new estimate $\hat{\theta}^{new}$. These steps will be described in detail below.

5.1.1 Expectation (E) Step

In the (E) step, an expectation of the log of joint likelihood is computed under the assumption of some known, initial value of $\hat{\theta}^{old}$. The expectation takes a form of a function of the bias vector θ, given by

$$Q(\theta, \hat{\theta}^{old}) = E_\theta \left[\log p(X_K, Y_K|\theta) \middle| Y_K, \hat{\theta}^{old} \right], \tag{5.2}$$

or equivalently

$$Q(\theta, \hat{\theta}^{old}) = \int \log p(X_K, Y_K|\theta) p(X_K|Y_K, \hat{\theta}^{old}) dX_K. \tag{5.3}$$

For the problem of calibration, considered in this thesis, a posterior for the set of states $p(X_K|Y_k, \hat{\theta}^{old})$ is obtained using a smoothing algorithm (Rauch-Tung-Striebel smoother in a linear case or particle smoother for the nonlinear case) based on the set of observations Y_K, when using a fixed value of the bias vector $\hat{\theta}^{old}$.

Algorithm 1 Expectation maximization (EM) algorithm

1. INITIALIZATION:
   - Initialize algorithm at iteration $n = 0$ with some initial value of the parameter vector $\hat{\theta}^{(n)} = \hat{\theta}^0$
2. For $n = 1 \ldots N$:
   - EXPECTATION (E) STEP:
     - Assign $\hat{\theta}^{old} = \hat{\theta}^{(n-1)}$
     - Compute $Q(\theta, \hat{\theta}^{old})$ as in (5.3)
   - MAXIMIZATION (M) STEP:
     - Obtain new estimate of θ as in (5.4)
     - Assign new estimate so $\hat{\theta}^{(n)} = \hat{\theta}^{new}$
3. If ($n == N$) or algorithm has converged: end iterations; otherwise: increment ($n = n + 1$) and continue.

5.1.2 Maximization (M) Step

In the maximization step a new value of $\hat{\theta}$ is obtained by maximizing the expectation (computed in the previous step), according to

$$\hat{\theta}^{new} = \arg\max_{\theta} Q(\theta, \hat{\theta}^{old}). \tag{5.4}$$

It is worthwhile to notice an important property, that was briefly mentioned earlier. Since in the (E) step one computes the expectation of a logarithm of the joint distribution, then in case when $p(X_K, Y_K|\theta)$ comprises a member of the exponential family the logarithm will cancel the exponential. In that case the maximization in (M) step will be much simpler than a maximization of incomplete data log likelihood $p(Y_K|\theta)$.

Another significant property of the expectation maximization algorithm is a fact that it has a guaranteed convergence to (at least local) optimum. This property is proven in Dempster et al. (1977) and Bishop (2006).

The iterative procedure of the expectation maximization algorithm is presented in Algorithm 1.

5.1.3 Computation of the MAP Estimate

In case the prior for parameter vector is given, $p(\theta)$, it is possible to use the expectation maximization algorithm to maximize the posterior $p(\theta|Y_K)$. Using a fact that we can decompose the logarithm of the posterior, it can be rewritten as

$$\log p(\theta|Y_K) = \log p(Y_K|\theta) + \log p(\theta) - \log p(Y_K), \tag{5.5}$$

by using the Bayes' rule and the properties of the logarithm. Since $p(Y_K)$ is a constant, it can be easily shown (Bishop (2006)), that a new $Q(\theta, \hat{\theta}^{old})$ function computed in the (E) step, for the maximum a posteriori estimate takes a form of

$$Q_{\text{MAP}}(\theta, \hat{\theta}^{old}) = Q(\theta, \hat{\theta}^{old}) + \log p(\theta). \tag{5.6}$$

In order to obtain a new MAP estimate of the parameter vector, one must maximize the expectation computed above with respect to θ. Hence

$$\hat{\theta}^{new} = \arg\max_{\theta} Q_{\text{MAP}}(\theta, \hat{\theta}^{old}). \tag{5.7}$$

5.2 Evidence Approximation (EA) Algorithm

When introducing a prior to the problem of maximum likelihood estimation, there arises a problem of assigning correct values of the hyper-parameters that govern the distribution. For the Gaussian prior, these parameters correspond to a precision (an inverse of a standard deviation) or to a scale parameter for the Laplace prior. For more details see Appendix A. In most cases the correct value of these parameters is unknown and needs to be determined from the measurement data.

This section briefly describes concepts underlying the evidence approximation method, introduced in MacKay (1992). It is a two step Bayesian inference method, where the Occam's razor is used to find the hyper-parameters. As a result a good balance between model complexity and fit to the measurement data is achieved.

The method is also known as Type-II Bayes (Berger, 1985) or sparse Bayesian learning (Wipf and Rao, 2004), and is as well the core idea underlying the relevance vector machines algorithm (Tipping, 2001).

5.2.1 Two Levels of Bayesian Inference

The idea underlying the method comes directly from the data modeling area, where one considers different models $\mathcal{M}_i$. Each i-th model is assumed to have a set of corresponding parameters θ and is defined by two probability distributions: a prior for parameters $p(\theta|\mathcal{M}_i)$ and a predictive distribution $p(D|\theta, \mathcal{M}_i)$.

The process of Bayesian modeling contains two inference levels. In the first step different models are fit to the observation data set D. Next, in the second step those models have to be evaluated and ranked in terms of how well they describe the particular data set.

In the first level of inference one assumes the model $\mathcal{M}_i$ is correct and fits that model to the data set D. Using Bayes' rule, the posterior for the parameter vector is given by

$$p(\theta|D, \mathcal{M}_i) = \frac{p(D|\theta, \mathcal{M}_i)p(\theta|\mathcal{M}_i)}{p(D|\mathcal{M}_i)}. \tag{5.8}$$

Second level of the inference is the model comparison, where one wants to find out which model is the most reliable given the data set. Using Bayes' again, the posterior probability for each of the models is proportional to

$$p(\mathcal{M}_i|D) \propto p(D|\mathcal{M}_i)p(\mathcal{M}_i). \tag{5.9}$$

Only the first term on the right hand side of the above equation depends on the

data set. The term is called the evidence for the model $\mathcal{M}_i$, and was seen above as normalizing constant in (5.8). Having the same prior $p(\mathcal{M}_i)$ for each of the models, to rank different models it is only necessary to evaluate the evidence. The evidence is computed by integrating out the set of the parameters

$$p(D|\mathcal{M}_i) = \int p(D|\theta, \mathcal{M}_i)p(\theta|\mathcal{M}_i)d\theta. \tag{5.10}$$

Interesting feature of the evidence is that it can be evaluated for the full spectrum of models, including parametric and non-parametric models. It is therefore a very useful and portable measure of the model preference.

5.2.2 Occam's Razor

Under an assumption that the posterior for parameters $p(\theta|D, \mathcal{M}_i)$ in (5.8) is sharply peaked around its most probable value $\hat{\theta}$, the evidence can be approximated as

$$p(D|\mathcal{M}_i) \simeq p(D|\hat{\theta}, \mathcal{M}_i)p(\hat{\theta}|\mathcal{M}_i)\Delta\theta, \tag{5.11}$$

where $p(D|\hat{\theta}, \mathcal{M}_i)p(\hat{\theta}|\mathcal{M}_i)$ corresponds to a height of the peak of the integrand and $\Delta\theta$ corresponds to its width. Or in other words

$$\text{Evidence} \simeq \text{Best fit likelihood} \times \text{Occam factor}. \tag{5.12}$$

The Occam factor is here interpreted as a penalty for the model complexity. In the process of evidence maximization the best model, corresponding to a trade-off between maximizing the fit to the data and minimizing its complexity, can be chosen. A full and exhaustive description and analysis of the evidence procedure and Occam's razor can be found in MacKay (1992).

5.2.3 Application to the Calibration Problem

The approach presented above can be easily applied to the problem of sensor network calibration. Analogously to the problem of maximum a posteriori estimation, it is assumed that the dataset $D = \{X_K, Y_K\}$ contains both set of sensor measurements Y_K and corresponding target states X_K for all time instances $k = 1 \dots K$. The model is determined by a set of (unknown) bias parameters θ, that we wish to estimate together with corresponding hyper-parameters α of the prior $p(\theta|\alpha)$. In this application different sets of hyper-parameters are considered as different models $\mathcal{M}_i$.

In the first level of inference model is fit to the data under the assumption, that the set of hyper-parameters is known ($\alpha = \hat{\alpha}$). By applying the Bayes' formula, and at the same time skipping X_K for clarity in the notation, the posterior for θ is given as

$$p(\theta|Y_K, \hat{\alpha}) = \left.\frac{p(Y_K|\theta, \alpha)p(\theta|\alpha)}{p(Y_K|\alpha)}\right|_{\alpha = \hat{\alpha}}. \tag{5.13}$$

Instead of integrating out the unknown set of hyper-parameters from

$$p(\theta|Y_K) = \int p(\theta|Y_K, \alpha)p(\alpha|Y_K)d\alpha, \tag{5.14}$$

as it should be done properly in a fully Bayesian way, the following approximation was used

$$p(\theta|Y_K) \simeq p(\theta|Y_K, \alpha)\Big|_{\alpha = \hat{\alpha}}. \tag{5.15}$$

It is allowed under an assumption that posterior for α is sharply peaked around the estimated value of $\hat{\alpha}$, and thus $p(\alpha|Y_K) \approx \delta(\alpha - \hat{\alpha})$.

In the second level of inference one aims to find the value of unknown hyper-parameter α, that maximizes the evidence function. Here the Occam's razor is applied to tune the parameters in a way, that the value of α is as small enough to fit the data to the model, but not too small (to avoid the over-fitting).

Using Bayes' formula again, analogously to the previous step, posterior for α is given by

$$p(\alpha|Y_K) = \frac{p(Y_K|\alpha)p(\alpha)}{p(Y_K)}. \tag{5.16}$$

Assuming a flat prior $p(\alpha)$ for the hyper-parameters, the only factor that counts is the evidence $p(Y_K|\alpha)$, the normalizing factor in (5.13). The evidence is obtained by integrating out θ from

$$p(Y_K|\alpha) = \int p(Y_K|\theta, \alpha)p(\theta|\alpha)d\theta. \tag{5.17}$$

Finally in order to obtain a value for the hyper-parameters, the evidence function is maximized with respect to α

$$\hat{\alpha} = \arg\max_{\alpha} p(Y_K|\alpha). \tag{5.18}$$

The algorithm takes an iterative form, where one switches between two levels of inference. In the first step, assuming some initial value of $\hat{\alpha}$, a new value of $\hat{\theta}$ is estimated. It is then used in the next step to compute a new value of $\hat{\alpha}$ by maximizing the evidence. The procedure is repeated until the convergence.

The iterative procedure of the evidence approximation is presented in the Algorithm 2.

5.2.4 Application of the EM Algorithm

Taking another look at the evidence function presented in (5.17), it is easily seen that θ can be considered as a latent variable. Thus it should be possible to apply the EM algorithm to iteratively estimate $\hat{\alpha}$. It is a fact indeed, and according to Algorithm 1, in the expectation step one needs to compute the expectation of a log of joint likelihood $p(Y_K, \theta|\alpha)$, using some initial value of $\hat{\alpha}^{old}$. This is done as

$$Q(\alpha, \hat{\alpha}^{old}) = \int p(Y_K, \theta|\alpha)p(\theta|Y_K, \hat{\alpha}^{old})d\theta, \tag{5.19}$$

Algorithm 2 Evidence approximation (EA) algorithm

1. INITIALIZATION:
   - Initialize algorithm at iteration $n = 0$ with some initial values of the parameter vector $\hat{\theta}^{(n)} = \hat{\theta}^0$ and the hyper-parameters $\hat{\alpha}^{(n)} = \hat{\alpha}^0$
2. For $n = 1 \ldots N$:
   - 1ST LEVEL OF INFERENCE:
     - Assign $\hat{\alpha}^{old} = \hat{\alpha}^{(n-1)}$
     - Compute $p(\theta|Y_K, \hat{\alpha}^{old})$ according to the (5.13) together with the point estimate $\hat{\theta}^{new}$
     - Assign $\hat{\theta}^{(n)} = \hat{\theta}^{new}$
   - 2ND LEVEL OF INFERENCE:
     - Compute the evidence function $p(Y_K|\alpha)$, using results from the previous step of inference, as in (5.17)
     - Maximize the evidence to obtain a new value $\hat{\alpha}^{new}$
     - Assign the new hyper-parameter estimate so $\hat{\alpha}^{(n)} = \hat{\alpha}^{new}$
3. If $(n == N)$ or algorithm has converged: end iterations;
   otherwise: increment $(n = n + 1)$ and continue.

where $p(\theta|Y_K, \hat{\alpha})$ is computed as in (5.13). Next, in maximization step the $Q(\alpha, \hat{\alpha}^{old})$ is maximized with respect to α in order to obtain new estimate $\hat{\alpha}$ as

$$\hat{\alpha}^{new} = \arg\max_{\alpha} Q(\alpha, \hat{\alpha}^{old}). \tag{5.20}$$

The procedure is repeated iteratively until convergence.

Application of the EM to that problem has a very useful practical consequence – it is then natural to extend the problem to a case with one extra latent variable – unknown set of states X_K of the observed target. The problem takes then a form of maximization of the evidence function obtained through the double integration

$$p(Y_K|\alpha) = \int\int p(X_K, Y_K, \theta|\alpha) dX_K d\theta. \tag{5.21}$$

The EM applies here in a natural way by decomposing the problem into two subsequent expectation steps, where the first is used to compute new values of $\hat{\alpha}$. A new $Q(\cdot)$ function can be stated as

$$Q(\alpha, \hat{\alpha}^{old}) = \int\int \ln p(X_K, Y_K, \theta|\alpha) p(X_K, \theta|Y_K, \hat{\alpha}^{old}) dX_K d\theta. \tag{5.22}$$

In order to solve the problem, as it is in detail described in Paper C, it is further necessary to compute the posterior for θ. It is due to the fact that (5.22) can be decomposed into

$$Q(\alpha, \hat{\alpha}^{old}) = \int g(\alpha) \int p(X_K, \theta|Y_K, \hat{\alpha}^{old}) dX_K d\theta + const, \tag{5.23}$$

where $g(\alpha)$ is some function of α, written in a short form for simplicity. Then, the

inner integral is equivalent to computing the aforementioned posterior for θ as

$$p(\theta|Y_K, \hat{\alpha}^{old}) = \int p(X_K, \theta|Y_K, \hat{\alpha}^{old}) dX_K. \tag{5.24}$$

Analogously to the case in Section 5.1, this problem is a problem of computing the posterior for bias vector in a presence of the latent variable X_K – the unknown target state. Thus applying the EM again, the expectation step yields

$$Q_{\text{MAP}}(\theta, \hat{\theta}^{old}) = \int \ln p(X_K, \theta|Y_K, \hat{\alpha}^{old}) p(X_K|Y_K, \hat{\theta}^{old}, \hat{\alpha}^{old}) dX_K + \ln p(\theta|\hat{\alpha}), \tag{5.25}$$

where the posterior for X_K can be computed using one of the available filtering/smoothing procedures.

After providing both expectations, the corresponding maximization steps are performed. One computes there a new value for both parameters θ and hyper-parameters α respectively. Both steps of Bayesian inference using the EM procedure are performed iteratively until convergence.

The iterative procedure of evidence approximation using the EM is presented in Algorithm 3. A detailed description of the process with two latent variables, followed by a further discussion, is presented in Paper C.

5.3 Summary

This chapter handles the advanced problems in the parameter estimation – the case of unknown (so called latent) variables and unknown priors.

In the first section, the application of the expectation maximization (EM) iterative procedure, to solve the problem of simultaneous tracking and parameter estimation is described. The underlying idea is further extended in Paper A, where the calibration problem is solved with a help of the road constraints used to improve the performance of filtering and smoothing.

In further sections the problem of estimating the hyper-parameters of the prior was described in the evidence maximization framework. The algorithm was introduced and it was briefly shown how to apply it to the problem of calibration of a sensor network. Further reading on that problem can be found in Paper B, which provides also an ad-hoc solution to the calibration problem where both target states and the hyper-parameters are unknown.

Finally, a double-looped iterative procedure for maximization of the evidence function using the EM was introduced. Further reading, together with a complete derivation and examples of application of the algorithm, can be found in Paper C.

The resulting algorithm is relatively robust to the case where the number of measurements is low (due to the application of the prior) and at the same time allows the user to simultaneously estimate both, the bias parameters and the unknown

Algorithm 3 Sparse calibration algorithm using EM for evidence approximation

1. Initialize algorithm at iteration $n = 0$ with some initial values of the parameters $\hat{\theta}^{(n)} = \theta^0$ and the hyper-parameters $\hat{\alpha}^{(n)} = \alpha^0$.
2. For $n = 1 \ldots N$:
   - Initialize algorithm with $\hat{\alpha} = \hat{\alpha}^{(n-1)}$

   FIRST LEVEL OF INFERENCE:
   - Initialize the EM algorithm at $j = 0$ with the initial values $\hat{\theta}^{(j)} = \hat{\theta}^{(n-1)}$
   - For $j = 1 \ldots J$:

     EXPECTATION (E_1) STEP:
     - Assign $\hat{\theta}^{old} = \hat{\theta}^{(j-1)}$
     - Obtain $Q_{\text{MAP}}(\theta, \hat{\theta}^{old})$ function according to (5.25)

     MAXIMIZATION (M_1) STEP:
     - Obtain a new estimate of θ that maximizes $Q_{\text{MAP}}(\theta, \hat{\theta}^{old})$
     - Assign the new estimate, so $\hat{\theta}^{(j)} = \hat{\theta}^{new}$
   - If $(j == J)$ assign $\hat{\theta}^{(n)} = \hat{\theta}^{(J)}$ and go to (E_2);
     otherwise: increment $(j = j + 1)$ and go to (E_1).

   SECOND LEVEL OF INFERENCE:
   - EXPECTATION (E_2) STEP:
     - Assign $\hat{\alpha}^{old} = \hat{\alpha}^{(n-1)}$
     - Compute $Q(\alpha, \hat{\alpha}^{old})$ as in (5.22)
   - MAXIMIZATION (M_2) STEP:
     - Obtain a new estimate of α that maximizes $Q(\alpha, \hat{\alpha}^{old})$
     - Assign the new estimate, so $\hat{\alpha}^{(n)} = \hat{\alpha}^{new}$
3. If $(n == N)$ or algorithm has converged: end iterations;
   otherwise: increment $(n = n + 1)$ and continue.

states of the target. Form of the evidence function implies Occam's razor, that provides a good balance between model complexity, represented in this case by number of non-zero bias parameters, and a fit to the measurement data. The sparsity is obtained through application of shrinkage priors, with the hyper-parameters estimated during the evidence maximization process.

6

Concluding Remarks

6.1 Conclusions

The below conclusions have been in a large part previously presented at the end of each of the corresponding chapters. In this section some of them will be repeated for the completeness of the final summary.

The thesis begins with a brief introduction to the problem of calibration of sensor networks, which is a main subject of this thesis. Problem is presented together with a list of contributions that are all contained in Part II. Next chapters describe the sensor networks in detail together with an introduction to both basic and more advanced calibration techniques.

After the introduction of the problem of calibration in the first chapter, Chapter 2 describes what the sensor network is in detail. Different sensor models are presented together with typical biases one might expect and sample measurements.

Having defined the measurement model, the calibration problem is stated in Chapter 3. The impact miscalibration might have on the performance of state estimation is presented, together with the analysis of the sensitivity to different bias values.

In Chapter 4 a basic estimation methods were introduced. Regularization concept was presented and it was shown how the ℓ_1 and ℓ_2 regularization correspond to application of different priors (Laplace and Gaussian respectively) in MAP estimation. End of the chapter contains a discussion about the observability problem and the CRLB.

In order to provide necessary background, a basic concept of Bayesian filtering and smoothing was introduced in Chapter 4. It was followed by a basic solution

to simultaneous tracking and network calibration problem through augmentation of the target state vector with bias vector.

Chapter 5 describes the problem of calibration with targets of opportunity together with an introduction to sparse calibration. Expectation maximization (EM) technique is applied to solve the simultaneous problem of the state and the parameter estimation, with further details provided in Paper A. Next introduced problem is the estimation of the hyper-parameters of the prior. The evidence approximation (EA) method is applied together with the so-called shrinkage priors, where thanks to the Occam's razor a good trade-off between model complexity (number of non-zero bias parameters) and fit to the measurement data is obtained.

Finally, a problem of application of the evidence method in case of unknown set of the target states is considered. Thanks to the application of the EM algorithm, both the filtering and the sparse parameter estimation using Occam's razor can be done simultaneously. Further details on the EA can be found in Paper B and the extension using EM is in detail described in Paper C.

6.2 Future Work

Further research regarding the calibration problem should investigate in more detail the simultaneous state and parameter estimation. More effort should be put on developing methods for non-linear and non-Gaussian cases, with a direct attention towards numerical methods. There is a great room for application of Markov chain Monte Carlo methods or Gaussian Process techniques in order to solve some of the problems in the area of calibration.

Further focus should be also put on the different particle filtering methods, including implementation using the road and the terrain constraints as a remarkable source of extra information. As was shown in Paper A, the application of simple road constraints can improve the filtering results and remove some degrees of freedom. It makes the problem much better determined – especially in the terms of a small number of measurements.

The limited number of measurements is a problem by itself. The application of regularization/priors helps to solve this difficult problem at least within some constraints. It should be thus reasonable to further investigate different regularization methods with respect to the calibration problem.

Last area that was insufficiently explored is the application of various machine learning methods. There still exists a vast number of algorithms that can be investigated in terms of possible application in the area of data fusion, in other to improve currently used techniques and, hopefully, provide some useful results.

Appendix

A

Probability Distributions

The chapter provides the basic knowledge about the probability distributions used in the rest of the thesis.

A.1 Gaussian Distribution

The Gaussian (also called the Normal) distribution is one of the basic distributions used in statistics. It belongs to the class of exponential distributions and in a one dimensional case it has a probability density function given as

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}, \tag{A.1}$$

where x is a scalar, μ represents a mean of the distribution and σ is a standard deviation (σ^2 thus corresponds to a variance). An inverts of the variance is called the precision ($\alpha = 1/\sigma^2$). The random variable x with a Gaussian distribution is considered to be normally distributed. The distribution for different parameter values is presented on Figure A.1.

In a multidimensional case the vector x of a size $N \times 1$ is distributed with a generalized one-dimensional distribution called a multivariate Gaussian (or Normal) distribution. It is expressed by

$$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} e^{-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)^T}, \tag{A.2}$$

where μ is a mean of size $N \times 1$ and σ is a $N \times N$ covariance matrix. The inverse of the covariance matrix is called a precision matrix ($A = \Sigma^{-1}$). A two dimensional example distribution is presented on Figure A.2.

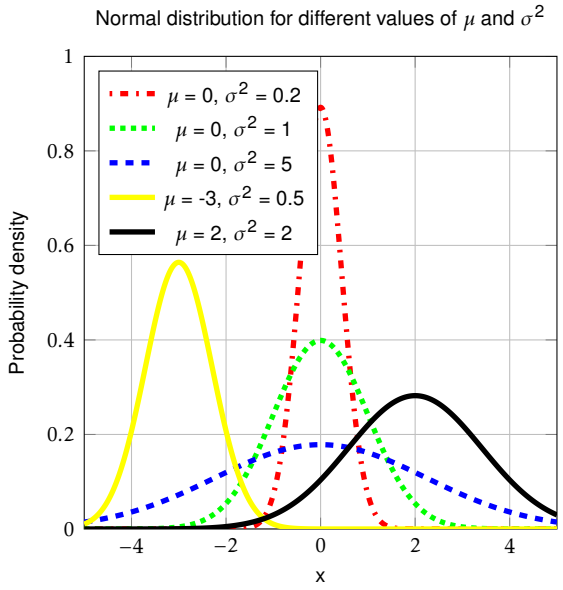


Figure A.1: *One dimensional Gaussian distribution.*

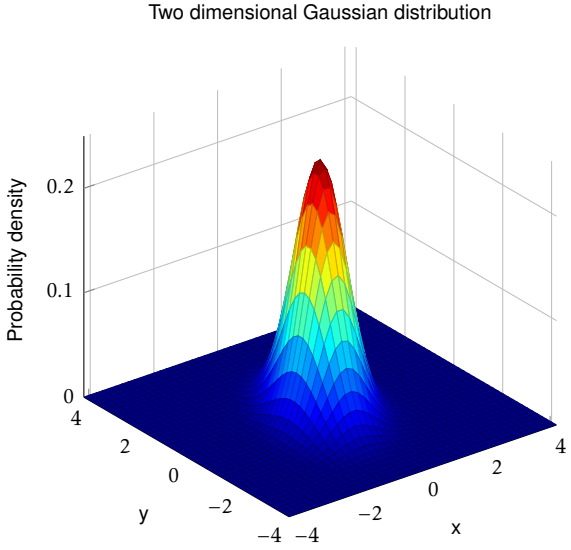


Figure A.2: *Two dimensional Gaussian distribution.*

A.2 Laplace Distribution

Another type of the exponential distribution considered in this thesis is a Laplace distribution. A random variable x has the Laplacian distribution with the proba-

bility density function given by

$$\mathcal{L}(x|\mu, b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}, \tag{A.3}$$

where μ is a location parameter and a scale parameter $b \geq 0$. The distribution for different parameter values is presented on Figure A.3.

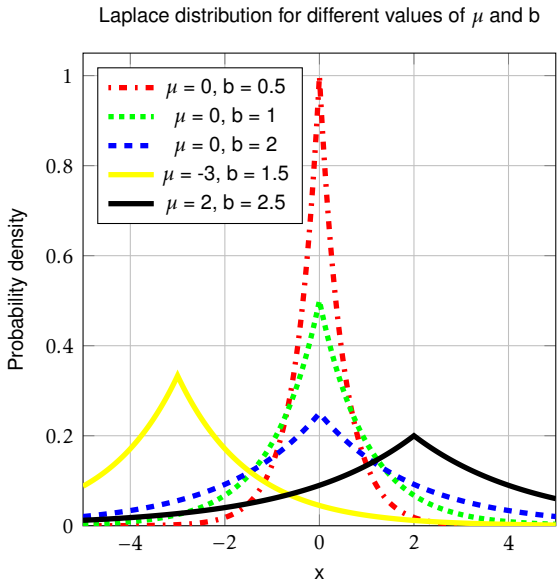


Figure A.3: *One dimensional Laplace distribution.*

It is worthwhile to notice that, in opposition to the Gaussian case, there does not exist a natural unique multivariate Laplace distribution. There appear few approaches to solving the problem or stating the multivariate Laplace distributions. For example in Anderson (1992), author presents the distribution as a specific multivariate Linnik distribution. On the other hand, in Eltoft et al. (2006) the distribution is presented as a multivariate scale mixture of Gaussian models, using an exponential prior for the scale factor. Thus differentiation bases on the assumption of which properties we want the distribution to carry, and which are less important and thus can be neglected.

Bibliography

Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *Communications magazine, IEEE*, 40(8):102–114, 2002.

Dale N Anderson. A multivariate Linnik distribution. *Statistics & probability letters*, 14(4):333–336, 1992.

Christophe Andrieu and Arnaud Doucet. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):827–836, 2002.

M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.

Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

James O Berger. *Statistical decision theory and Bayesian analysis*. Springer, 1985.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

Mikael Brännström, Ron Lennartsson, Andris Lauberts, Hans Habberstad, Erland Jungert, and Martin Holmberg. Distributed data fusion in a ground sensor network. In *Proc 7th International Conference on Information Fusion*. Citeseer, 2004.

Yoram Bresler. Two-filter formulae for discrete-time non-linear Bayesian smoothing. *International Journal of Control*, 43(2):629–641, 1986.

Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

Chee-Yee Chong and Srikanta P Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

S Dhar. Application of a recursive method for registration error correction in tracking with multiple sensors. In *American Control Conference, 1993*, pages 869–874. IEEE, 1993.

Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12:656–704, 2009.

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3): 197–208, 2000.

Torbjørn Eltoft, Taesu Kim, and Te-Won Lee. On the multivariate Laplace distribution. *Signal Processing Letters, IEEE*, 13(5):300–303, 2006.

F. Gustafsson. *Statistical Sensor Fusion*. Utbildningshuset/Studentlitteratur, 2012. ISBN 9789144077321.

Fredrik Gustafsson. Particle filter theory and practice with positioning applications. *Aerospace and Electronic Systems Magazine, IEEE*, 25(7):53–82, 2010.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

M Ignagni. An alternate derivation and extension of Friendland's two-stage Kalman estimator. *Automatic Control, IEEE Transactions on*, 26(3):746–750, 1981.

David JC MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

Geoffrey J McLachlan, Thriyambakam Krishnan, and See Ket NG. The EM algorithm. Technical report, Papers/Humboldt-Universität Berlin, Center for Applied Statistics and Economics (CASE), 2004.

P-J Nordlund and Fredrik Gustafsson. Marginalized particle filter for accurate and reliable terrain-aided navigation. *Aerospace and Electronic Systems, IEEE Transactions on*, 45(4):1385–1399, 2009.

Henrik Ohlsson. *Regularization for Sparseness and Smoothness: Applications in System Identification and Signal Processing*. PhD thesis, Linköping, 2010.

Branko Ristic, Sanjeev Arulampalm, and Neil James Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.

Egils Sviestins. Online bias estimation for multisensor tracking. In *Information, Decision and Control, 1999. IDC 99. Proceedings. 1999*, pages 221–226. IEEE, 1999.

M. Syldatk and F. Gustafsson. Expectation maximization algorithm for simultaneous tracking and sparse calibration of sensor networks. In *IEEE Transactions on Signal Processing (TSP)*. To be submitted.

M. Syldatk and F. Gustafsson. Simultaneous tracking and sparse calibration in ground sensor networks using evidence approximation. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

M. Syldatk, E. Sviestins, and F. Gustafsson. Expectation maximization algorithm for calibration of ground sensor networks using a road constrained particle filter. In *Proceedings of the 15th International Conference on Information Fusion (FUSION)*, pages 771–778, Singapore, July 2012.

Michael E Tipping. Sparse Bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

Jaco Vermaak, Simon Maskell, and Mark Briers. Online sensor registration. In *Aerospace Conference, 2005 IEEE*, pages 2117–2125. IEEE, 2005.

David P Wipf and Bhaskar D Rao. Sparse Bayesian learning for basis selection. *Signal Processing, IEEE Transactions on*, 52(8):2153–2164, 2004.

Part II

Publications

Paper A

Expectation Maximization Algorithm for Calibration of Ground Sensor Networks using a Road Constrained Particle Filter

Authors: Marek Syldatk, Fredrik Gustafsson and Egils Sviestins

Edited version of the paper:

M. Syldatk, E. Sviestins, and F. Gustafsson. Expectation maximization algorithm for calibration of ground sensor networks using a road constrained particle filter. In *Proceedings of the 15th International Conference on Information Fusion (FUSION)*, pages 771–778, Singapore, July 2012.

Expectation Maximization Algorithm for Calibration of Ground Sensor Networks using a Road Constrained Particle Filter

Marek Syldatk, Fredrik Gustafsson * and Egils Sviestins **

*Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
marek.syldatk@liu.se,
fredrik.gustafsson@liu.se

**Data Fusion, Command and Control Systems
Security and Defence Solution
Saab AB, Järfälla, Sweden
egils.sviestins@saabgroup.com

Abstract

Target tracking in ground sensor networks requires an accurate calibration of sensor positions and orientations, as well as sensor offsets and scale errors. We present a calibration algorithm based on the EM (expectation maximization) algorithm, where the particle filter is used for target tracking and a non-linear least squares estimator is used for estimation of the calibration parameters. The proposed algorithm is very simple to use in practice, since no ground truth of the target position and time synchronization are needed. In that way, opportunistic targets can also be used for calibration. For road-bound targets, a road-constrained particle filter is used to increase the performance. Tests on real data shows that a sensor position accuracy of a couple of meters is obtained from only one passing target.

1 Introduction

Bias estimation (encompassing the concepts sensor calibration, registration, localization and alignment) is a crucial element of multisensor tracking. Reliable tracking requires the sensor data to be aligned and consistent, otherwise there is an obvious risk for degraded performance or even ghost tracks.

There are many previous works dealing with bias estimation (for example Sviestins (1999) or Dhar (1993)). Here we focus on methods applicable to a ground sensor network. The different biases, here denoted bias parameters, may include location errors, orientation errors, range measurements etc., depending on sensor type.

There are two main groups of techniques for calibrating the sensors: using reference targets, or using targets of opportunity. Both can be performed on-line or off-line.

In the ground sensor applications, the reference targets could typically be one or more vehicles carrying satellite navigation equipment, e.g. GPS. One must make sure that enough measurements are generated for each sensor with the target at different locations. Problems to consider are how to make the GPS data available to the fusion centre, and also the fact, that the GPS data may be corrupt, especially in an urban environment. Once the data has been collected, one can apply e.g. a maximum likelihood (ML) method, to estimate the bias parameters that make observations of the target fit the reference data best.

In case no reference targets can be used, one needs to track targets of opportunity for the purpose of bias estimation. This may however be hard, as tracking may not work well without having the bias estimates. One approach is to simultaneously estimate track states and biases, which can be accomplished by forming augmented state vectors that combine target state estimates together with bias estimates (Dhar, 1993). With a large number of targets and sensors this will hardly be tractable due to computational requirements etc. There are however suboptimal but very efficient techniques to decouple target state and bias estimation process (Ignagni (1981) or Sviestins (1999)), fully feasible for on-line operation.

This paper examines an *off-line* approach to bias estimation with targets of opportunity. As the number of measurements suitable for bias estimation can be quite low, it is desirable to find a method that uses them as effectively as possible. To this end, the method considered in this paper is based on expectation maximization (EM) (Schön et al., 2011), with additional performance enhancing features, in particular using road constraints together with particle filtering and smoothing. In EM, the entire set of measurements is processed iteratively to provide both state estimates and bias parameters (Li et al., 2010). This will be described in detail in Section 3 after the formal definition in Section 2. The techniques are developed further in Section 4 and Section 5 with particle filtering and smoothing for on-road and off-road motion. Finally, Section 6 provides experimental results.

2 Problem Definition

2.1 Sensor and Motion Models

A ground target is detected by a number of sensors, resulting in a set of observations $Y_K = \{y_k\}_{k=1}^K$ of a target states $X_K = \{x_k\}_{k=1}^K$, where y_k is a single measurement, or set of measurements stacked as a vector, and x_k is target state at time k. The measurements are affected by a number of bias parameters and by a measurement noise. All the bias parameters, for all sensors, are collected in a single vector θ.

The measurement model for each sensor is assumed to be a known, nonlinear function $h(x_k, \theta)$ of the state x_k, with measurement bias θ, and is defined as

$$y_k = h(x_k, \theta) + v_k, \tag{1}$$

where ν_k is an additive noise with known distribution $p_\nu(\cdot)$.

In general we assume that observed target moves according to a nonlinear motion model

$$x_{k+1} = f(x_k, \eta_{k+1}), \tag{2}$$

with motion model $f(\cdot)$ assumed to be known and with η_k being a process noise with known distribution $p_\eta(\cdot)$.

There is a number of sensor types used in ground target tracking, from which we can distinguish four basic types:

Linear range (TOA, time of arrival) The time τ of a signal arrival (transmission time) is measured, and by using the known speed v in the propagation media (speed of light/sound in air/water etc.) one can estimate range to the target. The measurement function is defined as

$$h_{TOA}(x_k, \theta) = |p_k - (s + \theta_p)| + \theta_r, \tag{3}$$

where p_k is target positions vector at time k, s is the sensor position vector, θ_p is the sensor position bias vector and θ_r is an additive range measurement bias.

Linear range difference (TDOA, time difference of arrival) The transmission time is here unknown, but by using two sensors one can estimate a relative time of arrival, thus relative distance to the target. The measurement function takes then a form

$$h_{TDOA}(x_k, \theta) = |p_k - (s^1 + \theta_p^1)| - |p_k - (s^2 + \theta_p^2)| + \theta_d, \tag{4}$$

where s^n is the n-th sensor position vector, θ_p^n is the n-th sensor position bias vector and θ_d is an additive range difference measurement bias.

Bearing measurements (DOA, difference of arrival) The sensor provides bearing (direction, angle) to the target, with measurement function

$$h_{DOA}(x_k, \theta) = \operatorname{atan2}\left(\frac{p_k^x - (s_x + \theta_x)}{p_k^x - (s_y + \theta_y)}\right) + \theta_\theta, \tag{5}$$

where p_k^x and p_k^y are target positions, s_x and s_y are sensor positions, θ_x and θ_y are sensor position biases in x and y direction respectively and θ_θ is called north alignment bias.

Logarithmic range (power measurements). The sensor measures the strength of a received signal, that can be of a different kind, e.g. acoustic or electromagnetic wave. The measurement function is then defined as

$$h_{RSS}(x_k, \theta) = \log(1 - \theta_G) + P_k^{LOG} - \beta \log(r), \tag{6}$$

where $r = |p_k - (s + \theta_p)|$ is the distance to the target, P_k^{LOG} is a logarithm of power emitted by the target at time k, β is called path loss constant and θ_G is a gain bias related to miscalibration of sensors.

The list of sensor models and bias parameters above is not exhaustive, but can

be considered as a good illustration of the calibration problem considered herein. Further, sensor modalities can be mixed arbitrarily in our framework. Also combined sensors can be included easily, such as a radar sensor, providing both range and bearing measurements.

2.2 General Estimation Framework

The estimation approach consists in finding the bias parameters θ that maximize the marginal likelihood

$$p_\theta(Y_K) = \int p_\theta(X_K, Y_K) dX_K, \tag{7}$$

where θ indicates dependence on bias parameters. The EM algorithm computes the maximum likelihood estimate by iteratively solving a filtering problem to get X_k given an estimate of the bias, and an estimation problem to get θ_k, given an estimate of X_k.

A general framework on the EM algorithm for estimating parameters in a nonlinear dynamic system is provided in Schön et al. (2011).

3 Expectation Maximization (EM) Algorithm

The ideas behind EM, as applicable to our problem, can be summarized as follows. It is an iterative method used for finding the maximum likelihood (ML) or maximum posterior (MAP) estimates of parameters, when a model depends on latent (hidden) values. In case of bias estimation, where we based on set of measurements Y_K want to estimate the bias vector θ, the latent variables are the sequence of state vectors X_K. The two steps described below are repeated until convergence. The first step is an expectation (E) step. Introducing the notation

$$\ln p_\theta(X_K, Y_K) = L_\theta(X_K, Y_K), \tag{8}$$

one computes the $Q(\theta, \hat{\theta})$, that is a minimum variance estimate Schön et al. (2011) of above log likelihood, made use of the available data set $\{y_k\}_1^K$ and an assumption $\hat{\theta}$ of the true value of bias vector θ. Function is defined as

$$Q(\theta, \hat{\theta}) = E_{\hat{\theta}}\{L_\theta(X_K, Y_K)|Y_K\} = \int L_\theta(X_K, Y_K) p_{\hat{\theta}}(X_K|Y_K) dX_K. \tag{9}$$

The core idea behind the above procedure is that it should be much easier to maximize the complete likelihood $p_\theta(X_K, Y_K)$ than $p_\theta(Y_K)$ in (7).

The second step is maximization (M) step, where one calculates new bias estimate

$$\hat{\theta}^{(n+1)} = \arg\max_\theta p_\theta(Y_K) = \arg\max_\theta Q(\theta, \hat{\theta}^{(n)}). \tag{10}$$

This vector is used as an input for the next iteration of the two steps. As a result, with further iterations, the algorithm delivers a sequence of estimates $\hat{\theta}^{(n)}$ that are increasingly, with each iteration, better approximations of the ML estimate. As an initial value θ_0 one can use zeros or any other data one may have available.

The EM algorithm is presented in Algorithm 1 and details of the expectation and maximization steps are given in the following sections.

Algorithm 1 EM algorithm

1. Set $n = 0$ and initialize $\hat{\theta}^{(n)}$
2. **(E)xpectation step:**
   Calculate: $Q(\theta, \hat{\theta}^{(n)})$
3. **(M)aximization step:**
   Calculate: $\hat{\theta}^{(n+1)} = \arg\max_\theta Q(\theta, \hat{\theta}^{(n)})$
4. Update $n = n + 1$ until convergence.

The important problem in the EM algorithm is evaluation of the distribution $p_{\hat{\theta}}(X_K|Y_K)$ in (9), which is a smoothing problem. It should be discussed a bit wider due to the fact that there exist multiple ways to obtain estimates of θ through EM.

In case of a linear measurement model and a linear motion model with normally distributed noise, one obtains a direct solution to the estimation problem by using a Kalman smoother in the expectation step together with weighted least squares (WLS) algorithm to obtain bias estimates in the maximization step. It is then a well known property of the EM algorithm that it will converge to a local maximum of the likelihood function.

In the nonlinear case we need to use approximative algorithms. One general solution to the nonlinear filtering problem is provided by the particle smoother, that can obtain an arbitrarily good approximation of the $Q(\theta, \hat{\theta}^{(n)})$ function. Then, one can use a gradient method to solve for θ (which is equivalent to nonlinear least squares (NLS) problem). This method in detail is presented in Schön et al. (2011).

Another way is to use linearization of the motion model and/or measurement model. With a linearized model, one can use the extended Kalman smoother to obtain smoothed state estimates (Li et al., 2010). Linearization allows for an iterative solution where the model is linearized at each new iterate.

In this paper, the method with linearized measurement function and particle smoother is used to obtain an approximation of $p_{\hat{\theta}}(X_K|Y_K)$.

3.1 Expectation Step

We now focus on how to calculate $Q(\theta, \hat{\theta}^{(n)})$ defined in (9). Using the Bayes' rule, the log likelihood (8) can be expressed as

$$L_\theta(X_K, Y_K) = \ln p_\theta(Y_K|X_K) + \ln p_\theta(X_K). \tag{11}$$

Actually, only the first term depends on the biases related to the measurement function, so under the assumption of known initial state distribution $p(x_1)$, the second term can be considered as a constant. Thus keeping only terms dependent on θ, and including other terms into the constant term, one can write (8) in an

extended form as

$$L_\theta(X_K, Y_K) = \sum_{k=1}^{K} \ln p_\theta(y_k|x_k) + const. \tag{12}$$

Assuming that measurements are obtained from true target states x_k through a known, nonlinear measurement function

$$y_k = h(x_k, \theta) + \nu_k, \tag{13}$$

defined in (1), with θ being the bias vector and ν_k assumed to follow a Normal distribution with a known covariance R, we then have

$$L_\theta(X_K, Y_K) = -\frac{1}{2}\sum_{k=1}^{K}(h^T(x_k, \theta)R^{-1}h(x_k, \theta) - 2y_k^T R^{-1}h(x_k, \theta)) + const, \tag{14}$$

where the terms independent of the bias θ were again considered as constants.

Due to its nonlinearity, the measurement function needs to be linearized in order to obtain a closed form solution for bias estimates. The best point for linearization is around the state estimate $\hat{x}_{k|K} = E_{\hat{\theta}}\{x_k|Y_K\}$. A motivation to use this value comes directly from the fact, that computing of $Q(\theta, \hat{\theta}^{(n)})$ requires calculation of this value. It is due to expectation over Y_k in (9), which requires computing $p_{\hat{\theta}}(X_K|Y_K)$. As will be seen below, linearization around the smoothed state estimate will simplify the final solution (by making some terms disappear).

In this paper, the particle filter together with a particle smoother will be applied to obtain smoothed state estimates. First, in the filtering step one obtains a set of particles $\{x_k^i\}_{i=1}^M$ together with their weights $\{w_k^i\}_{i=1}^M$, where M is the number of particles, that are used in the smoothing step to compute the smoothed estimate $\hat{x}_{k|K}$. Detailed description of these algorithms will be provided in Section 3 and 4 respectively.

To proceed, the measurement function is linearized around the smoothed estimate $\hat{x}_{k|K}$ and bias estimate $\hat{\theta}^{(n)}$ using a Taylor 1^{st} order expansion, resulting in

$$h(x_k, \theta) \approx h(\hat{x}_{k|K}, \hat{\theta}^{(n)}) + H_k^\theta(\theta - \hat{\theta}^{(n)}) + H_k^x(x_k - \hat{x}_{k|K}), \tag{15}$$

with Jacobians

$$H_k^x = \left.\frac{\partial h(x_k, \theta)}{\partial x_k}\right|_{\substack{x_k=\hat{x}_{k|K} \\ \theta=\hat{\theta}^{(n)}}}, \qquad H_k^\theta = \left.\frac{\partial h(x_k, \theta)}{\partial \theta}\right|_{\substack{x_k=\hat{x}_{k|K} \\ \theta=\hat{\theta}^{(n)}}}. \tag{16}$$

Substituting (15) into (14) and applying the expectation operator $E_{\hat{\theta}}\{\cdot|Y_K\}$ to (14), in order to calculate $Q(\theta, \hat{\theta}^{(n)})$ (cf. (9)), and by again throwing away terms that do not depend on θ, together with terms that are linear in $x_k - \hat{x}_{k|K}$, as their expectation value vanishes (as a direct result of linearization around smoothed state estimate). What remains is

$$Q(\theta, \hat{\theta}) = -\frac{1}{2}\sum_{k=1}^{K}(-2(y_k - h(\hat{x}_{k|K}, \hat{\theta}) + H_k^\theta\hat{\theta})^T R^{-1}H_k^\theta\theta + \theta^T(H_k^\theta)^T R^{-1}H_k^\theta\theta) + const. \tag{17}$$

3.2 Maximization Step

In the maximization step, we compute new bias estimate $\hat{\theta}^{(n+1)}$ from (10). The maximum can be found by solving

$$\frac{\partial Q(\theta, \hat{\theta}^{(n)})}{\partial \theta} = 0 \tag{18}$$

for θ and using the property of symmetry of measurement covariance matrix $R = R^T$. Because (17) has a form of $A\theta + \theta^T B\theta + const$, where

$$A = \sum_{k=1}^{K} (y_k - h(\hat{x}_{k|K}, \hat{\theta}) + H_k^{\theta}\hat{\theta})^T R^{-1} H_k^{\theta}, \tag{19}$$

$$B = \frac{1}{2} \sum_{k=1}^{K} (H_k^{\theta})^T R^{-1} H_k^{\theta}, \tag{20}$$

with θ as mentioned symmetric and $const$ independent of θ, the result is simply $\hat{\theta}^{(n+1)} = -B^{-1}A/2$, or explicitly

$$\hat{\theta}^{(n+1)} = \left(\sum_{k=1}^{K} (H_k^{\theta})^T R^{-1} H_k^{\theta} \right)^{-1} \left(\sum_{k=1}^{K} (H_k^{\theta})^T R^{-1} (y_k - h(\hat{x}_{k|K}, \hat{\theta}) + H_k^{\theta}\hat{\theta}) \right). \tag{21}$$

3.3 Comments on Observability and Convergence

For the maximization step to work it is necessary that θ is invertible (or rather, that $\hat{\theta}^{(n+1)} = -B^{-1}A/2$ is a well-conditioned problem). Typically this is not the case: there are too many degrees of freedom for the bias parameters. The freedom can be restricted in different ways: fixing certain parameters or using road constraints can be very helpful, but not always enough. There are also requirements on the set of measurements. If there are too few measurements, of if e.g. the target has not moved in the measurement set, it will in most cases also be impossible to find the parameters. Using priors is also another way to ensure observability.

The proof of a convergence of the EM algorithm can be found in literature in Bishop (2006) or Schön et al. (2011).

4 Particle Filtering

As mentioned in the previous section, to obtain bias estimates $\hat{\theta}^{(n)}$ through the EM algorithm, we require smoothed state estimates $\hat{x}_{k|K}$. To obtain those estimates, we will use the particle smoother. To be able to apply smoothing, we first need to run a particle filter, that will provide a set of particles, together with their weights, further used in a smoothing step. In this section, a simple SIR (sequential importance resampling) algorithm will be used to obtain the required particles and weights, followed by its modification with applied road constraints.

4.1 Particle Filter

In general, we assume that the observed target moves according to a nonlinear motion model defined in (2)

$$x_{k+1} = f(x_k, \eta_{k+1}), \tag{22}$$

with motion model $f(\cdot)$ assumed to be known and with η_k being a process noise with known distribution $p_\eta(\cdot)$. The measurement function used to obtain target observations was previously defined in (13). A simple SIR particle filter algorithm Gordon et al. (1993) Ristic et al. (2004) is presented in Algorithm 2.

Algorithm 2 SIR particle filter

1. **Initialize:** Set $k = 1$ and initialize particles
$$\{x_0^i\}_{i=1}^M \sim p(x_0) \tag{23}$$
2. **Prediction:** Predict the particles by drawing M i.i.d. samples according to
$$\tilde{x}_k^i \sim p(\tilde{x}_k | x_{k-1}^i), \qquad i = 1, \ldots, M \tag{24}$$
3. **Update:** Compute the importance weights $\{w_k^i\}_{i=1}^M$,
$$w_k^i = w(\tilde{x}_k^i) = \frac{p_\theta(y_k|\tilde{x}_k^i)}{\sum_{j=1}^M p_\theta(y_k|\tilde{x}_k^j)}, \qquad i = 1, \ldots, M \tag{25}$$
4. **Resampling:** For each $j = 1, \ldots, M$ draw a new particle x_k^i with replacement (resample) according to
$$P(x_k^j = \tilde{x}_k^i) = w_k^i, \qquad i = 1, \ldots, M \tag{26}$$
5. Increment $k = k + 1$ while $k < K$ and return to Step 2, otherwise terminate.

4.2 Road Constrained Particle Filter

When the target moves on a road, and a map of the road is available, it is natural to try to use this extra knowledge as a constraint in order to obtain more accurate estimates. In this section a road constrained particle filter (where the target moves only on the road map) will be described in detail.

There are many ways to apply road constraints to the particle filter. In this paper it will be assumed, that the road map is represented by a road network defined as J_{RN}. The road network is a set of definitions of straight segments together with a description of their attributes. Each segment is build of two points (a start point and an end point). Points common for three or more segments are considered to be junctions.

In general, the target state is defined in some coordinate system, let us call it global for a purpose of this paper. For the road constrained filtering there exists a need of using an extra coordinate system, that will describe target position in road coordinates. The state at time k in this system will be denoted as x_k^r, where r denotes road coordinate system.

The on-road state vector $x_k^r = [z_k \; l_k]^T$ is combined of two vectors: $z_k = [s_k \; v_k]^T$ describing a one-dimensional motion model, where s_k is a total distance traveled

on the road since $k = 1$, v_k is a speed, and l_k informs about the road segment the target is on, location on the segment and direction of movement.

The motion model for the on road case is, as in general case, also nonlinear and defined (Gustafsson et al., 2012) as

$$x^r_{k+1} = f^r(x^r_k, J_{RN}, \eta^r_{k+1}, \nu^r_{k+1}), \tag{27}$$

where η^r_k is a process noise with known distributions $p_{\eta^r}(\cdot)$ and ν^r_k is a discrete process noise, determining the choice of next road segment with known distributions $p_{\nu^r}(\cdot)$

Because the measurement function $h(x_k, \theta)$ is usually defined in global coordinates, there is a need to be able to convert target state from road to global coordinates. Thus we assume, there exists a transformation that allows exact conversion from global coordinates to road coordinates and opposite. Let $\Gamma^{r2g}(\cdot)$ be the function converting road coordinates to global coordinates and let $\Gamma^{g2r}(\cdot)$ be the transformation from global coordinates to road coordinates. In the on-road case, the measurement function (13) takes the form

$$y_k = h(\Gamma^{r2g}(x^r_k), \theta) + \nu_k. \tag{28}$$

The procedure for on road filtering is analogous to the one presented in Algorithm 2, but with a few modifications. In the initialization step, particles can be initialized directly on the road (in road coordinates) or projected onto road in the case when the initial distribution is only known in global coordinates. Then initial samples (particles) need to be projected using some known projection function. In general, equation (23) in Algorithm 2, for road constrained case, takes the form

$$\{x^{r,i}_0\}^M_{i=1} \sim p(x^r_0). \tag{29}$$

Next, in prediction step one needs to consider the on road motion model together with the probability of choosing one of the next road segments (in case of junctions or end of the road (Salmond et al., 2007)). The likelihood in (24), which we sample from in step (2) of Algorithm 2, is then

$$p(x^r_k|x^r_{k-1}) = p(z_k, l_k|z_{k-1}, l_{k-1}) = \frac{p(z_k, l_k, z_{k-1}, l_{k-1})}{p(z_{k-1}, l_{k-1})}, \tag{30}$$

where Bayes theorem was used. By using the property of independence of z_{k-1} and l_{k-1}, and Bayes theorem again, (30) can be rewritten as

$$\begin{aligned} p(x^r_k|x^r_{k-1}) &= \frac{p(l_k, l_{k-1}|z_k, z_{k-1})}{p(l_{k-1})} \frac{p(z_k, z_{k-1})}{p(z_{k-1})} \\ &= \frac{p(l_k|l_{k-1}, z_k, z_{k-1}) p(l_{k-1}|z_k, z_{k-1})}{p(l_{k-1})} p(z_k|z_{k-1}), \end{aligned} \tag{31}$$

Using the fact that $p(l_k|z_k, z_{k-1}) = p(l_k)$, terms from nominator and denominator disappear and we finally get

$$p(x^r_k|x^r_{k-1}) = p(z_k|z_{k-1}) p(l_k|l_{k-1}, z_k, z_{k-1}). \tag{32}$$

As was mentioned before, in the update step of Algorithm 2 there might be a need to convert particles from road to global coordinates. To calculate $p_\theta(y_k|\tilde{x}_k^i)$ we should use the coordinate conversion function, as in (28). Modified version

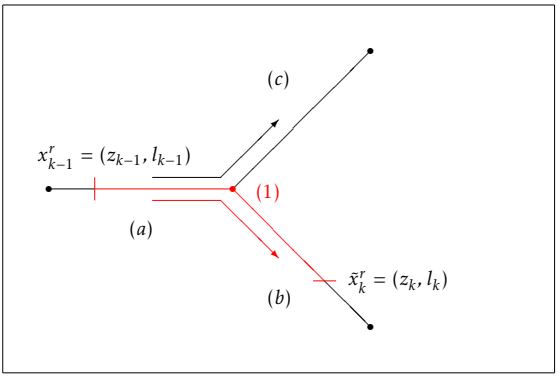


Figure 1: *Junction example.*

for road constrained particle filter is presented in Algorithm 3.

Algorithm 3 Road constrained particle filter

1. **Initialize:** Set $k = 1$ and initialize particles
$$\{x_0^{r,i}\}_{i=1}^M \sim p(x_0^r) \tag{33}$$
2. **Prediction:** Predict the particles by drawing M i.i.d. samples according to (32)
$$\tilde{x}_k^{r,i} \sim p(\tilde{x}_k^r|x_{k-1}^{r,i}), \qquad i = 1,\dots,M \tag{34}$$
3. **Update:** Compute the importance weights $\{w_k^i\}_{i=1}^M$,
$$w_k^i = w(\tilde{x}_k^{r,i}) = \frac{p_\theta(y_k|\Gamma(\tilde{x}_k^{r,i}))}{\sum_{j=1}^M p_\theta(y_k|\Gamma(\tilde{x}_k^{r,j}))}, \qquad i = 1,\dots,M \tag{35}$$
4. **Resampling:** For each $j = 1,\dots,M$ draw a new particle $x_k^{r,i}$ with replacement (resample) according to
$$P(x_k^{r,j} = \tilde{x}_k^{r,i}) = w_k^i, \qquad i = 1,\dots,M \tag{36}$$
5. Increment $k = k + 1$ while $k < K$ and return to Step 2, otherwise terminate.

Now we focus more deeply on sampling from the distribution $p(\tilde{x}_k^r|x_{k-1}^{r,i})$ in (34), with respect to known motion model and noise distribution. According to the first term in the equation, $p(\tilde{z}_k|z_{k-1}^i)$, we sample with respect to the on-road motion model. The second term depends on the road map, and is called junction selection likelihood.

The particle $x_{k-1}^{r,i}$ at time $k-1$ is located on a certain segment, and its location is described by the vectors z_{k-1} and l_{k-1} (as presented in Figure 1).

The first step is to sample a new $\tilde{z}_k$ from distribution $p(\tilde{z}_k|z_{k-1}^i)$. Having a new sample $\tilde{z}_k = [\tilde{s_k}\ \tilde{v_k}]^T$, we need to calculate the distance made by particle on road,

$d = \tilde{s}_k - s_{k-1}$, which is a difference between total distance s at time k and $k-1$. If the distance d is smaller or equal to the remaining distance to the junction (1) (end of a segment (a)), particle does not cross the junction and stays on segment (a) with probability 1, so $\tilde{l}_k = l_{k-1}$.

In case, when the distance d is larger than the remaining distance to the junction (1), the particle changes road segment. As we can see on the example in Figure 1, the particle has 2 possible paths to follow (red to segment (b) or black to segment (c)). In this paper, the distribution $p_{\nu^r}(\cdot)$, determining the choice of the next road segment, is assumed to be uniform, so particle changes its location to one of two possible segments ((b) or (c)) with an equal probability $\frac{1}{m}$, where m is a number of possible segments to move onto (in this example $m = 2$).

In case where at one time step the particle crosses more than one junction, we repeat the above procedure until the distance d is smaller than the remaining distance to the end of the segment the particle is on.

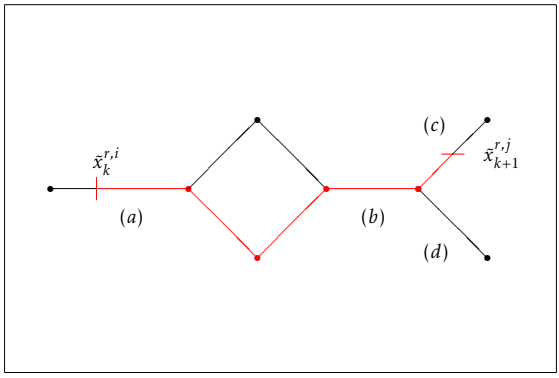


Figure 2: *Multiple junction example.*

5 Smoothing

As was mentioned before, to calculate smoothed state estimates we need to run first the particle filter and store all predicted particles $\{\tilde{x}_k^i\}_{i=1}^M$ and corresponding weights $\{w_k^i\}_{i=1}^M$ for all $k = 1, \ldots, K$ and then run Algorithm 4 (Schön et al. (2011)).

In case of road constrained algorithm, the above procedure needs to be modified, analogously to the filtering case. The main difference is in calculating the probability $p(\tilde{x}_{k+1}^j | \tilde{x}_k^i)$, between 2 particles, as in step (2) and (3) of Algorithm 4. In the road constrained case it follows, as defined in (32)

$$p(\tilde{x}_{k+1}^{r,j} | \tilde{x}_k^{r,i}) = p(\tilde{z}_{k+1}^j | \tilde{z}_k^i) p(\tilde{l}_{k+1}^j | \tilde{l}_k^i, \tilde{z}_{k+1}^j, \tilde{z}_k^i). \tag{40}$$

There are two main problems regarding evaluation of this probability.

Algorithm 4 Particle smoother

1. **Initialization:** Set filtered terminal weights $\{w_k^i\}$ to be initialized smoothed weights at time $k = K$ as
$$w_{K|K}^i = w_K^i, \qquad i = 1, \ldots, M \tag{37}$$
2. **Smoothing:** Use filtered weights $\{w_k^i\}$ and sets of stored particles $\{\tilde{x}_k^i, \tilde{x}_{k+1}^i\}_{i=1}^M$ to compute smoothed weights $\{w_{k|K}^i\}_{i=1}^M$ using formulas below:
$$w_{k|K}^i = w_K^i \sum_{j=1}^{M} w_{k+1|K}^j \frac{p(\tilde{x}_{k+1}^j|\tilde{x}_k^i)}{v_k^j} \tag{38}$$
where
$$v_k^j = \sum_{i=1}^{M} w_k^i p(\tilde{x}_{k+1}^j|\tilde{x}_k^i) \tag{39}$$
3. While $k > 0$ decrease $k = k - 1$ and return to Step (2), otherwise terminate.

The first term of equation (40) is directly related to the one-dimensional on-road motion model, as defined in (27). The second term is related to the road network. In a simple case, when particles $\tilde{x}_{k+1}^{r,j}$ and $\tilde{x}_k^{r,i}$ are located on the same segment, the second term is equal to 1. In case when particles are located on corresponding segments (problem analogous to the one presented in Figure 1), the probability is equal to $\frac{1}{m}$, where m is the number of possible segments the particle can choose.

In general case, when the on-road path between 2 particles crosses more than one junction (as presented in Figure 2), one needs to consider all the possible trajectories between two particles. As seen in the figure, particle moving from segment (a), through (b) to (c), can pick two possible trajectories. Thus, the second term in (40), for trajectories between (a) and (b) is equal to 1, and between (b) and (c) to $\frac{1}{m}$ with $m = 2$. That makes the junction selection probability between segment (a) and (c) equal to $1 \cdot \frac{1}{2} = \frac{1}{2}$.

A second problem is that the probability in (40) needs to be computed for each pair of particles in each smoothing step k, thus the complexity of one step is equal to M^2, where M is the number of particles. Because computation of the likelihood requires finding all possible paths between two points on-road, it is computationally demanding. This is considered as an important part to be improved in future.

Having the set of particles $\{\tilde{x}_k^i\}_{i=1}^M$ and their smoothed weights $\{w_{k|K}^i\}_{i=1}^M$, the minimum mean square error estimate of the target state can be computed as
$$\hat{x}_{k|K} = \sum_{i=1}^{M} w_{k|K}^i \tilde{x}_k^i, \tag{41}$$
and used to obtain bias estimate $\hat{\theta}^{(n)}$ through equation (21). Because the EM algorithm is, in general, defined in global coordinates, there is a need to convert estimates in case when road constraints are applied. In this paper particles in

road coordinates are converted first to global coordinates using the Γ^{r2g} mapping, and together with the smoothed weights are used to obtain the state estimate in global coordinates through

$$\hat{x}_{k|K} = \sum_{i=1}^{M} w_{k|K}^{i} \Gamma^{r2g}(\tilde{x}_{k}^{r,i}). \tag{42}$$

Because, in the above, we are using particles converted to global coordinates it might happen that the estimate will not be on-road. A natural remedy is to compute the particle that minimizes the mean square error,

$$\hat{x}_{k|K} = \arg\min_{\tilde{x}_k^j} \|\tilde{x}_k^i - \sum_{i=1}^{M} w_{k|K}^{i} \tilde{x}_k^i\|^2. \tag{43}$$

Since each particle is constrained to be on-road, this procedure will guarantee that also the estimate is on-road.

6 Experimental Results

The EM algorithm with applied road constraints will be compared with the version without constraints in the experiment, where a single target is moving on-road and is being observed by a number of sensors measuring (unknown) acoustic power emitted by the target. Evaluation will be performed on real data. In both versions of the EM algorithm, a constant velocity (CV) motion model (Gustafsson et al., 2012) was considered with an extra state representing logarithm of acoustic power P^{log} emitted by the target. The motion model in global coordinates, defined in (22), was represented by a two-dimensional CV model defined as

$$\begin{bmatrix} p_{k+1}^x \\ p_{k+1}^y \\ v_{k+1}^x \\ v_{k+1}^y \\ P_{k+1}^{log} \end{bmatrix} = \begin{bmatrix} I_2 & TI_2 & 0_{2\times 1} \\ 0_2 & I_2 & 0_{2\times 1} \\ 0_{1\times 2} & 0_{1\times 2} & 1 \end{bmatrix} \begin{bmatrix} p_k^x \\ p_k^y \\ v_k^x \\ v_k^y \\ P_k^{log} \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} I_2 & 0_{2\times 1} \\ TI_2 & 0_{2\times 1} \\ 0_{1\times 2} & 1 \end{bmatrix} \eta_{k+1}^{g}, \tag{44}$$

where $T = 1$, p_k^x and p_k^y represent x and y position at time k respectively, v_k^x and v_k^y represent x and y velocities at time k, P_k^{log} is the emitted acoustic power at time k and η_k is a three-dimensional Gaussian process noise with zero mean and covariance $Q = \mathrm{diag}([25\ 25\ 0.1])$.

The on-road motion model is defined as

$$\begin{bmatrix} s_{k+1} \\ v_{k+1} \\ P_{k+1}^{log} \\ l_{k+1} \end{bmatrix} = f^r \left(\begin{bmatrix} s_k \\ v_k \\ P_k^{log} \\ l_k \end{bmatrix}, J_{RN}, v_{k+1}^r, \eta_{k+1}^r \right), \tag{45}$$

where

$$\begin{bmatrix} s_{k+1} \\ v_{k+1} \\ P_k^{log} \end{bmatrix} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_k^r \\ v_k^r \\ P_k^{log} \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & 1 \end{bmatrix} \eta_{k+1}^r, \tag{46}$$

and where s_k and v_k represent a one-dimensional distance and velocity at time k respectively and η_k^r is a two-dimensional Gaussian process noise with zero mean and covariance $Q^r = \text{diag}([0.01\ 0.1])$ and ν_k^r is a discrete process noise with uniform distribution.

In this experiment, measurements are obtained from N identical acoustic sensors (microphones). The measurement function for the n-th sensor, as defined in (6), is

$$y_k^n = h_{RSS}^n(x_k, \theta^n) + \nu_k^n = log(1 - \theta_g^n) + P_k^{log} - \beta \log(r_n) + \nu_k^n, \tag{47}$$

where $r_n = \sqrt{((p_k^x - (s_x^n - \theta_x^n))^2 + (p_k^y - (s_y^n - \theta_y^n))^2)}$ is a distance from the sensor to the target, $\beta = 2$ is the path loss exponent, the bias vector for the n-th sensor is defined as $\theta^n = [\theta_x^n \quad \theta_y^n \quad \theta_g^n]^T$, where θ_x^n, θ_y^n and θ_g^n represent position biases in x and y sensor position and gain error respectively, s_x^n and s_y^n is a sensor position, p_k^x and p_k^y are positions of the target in global coordinates at time step k and ν_k^n represents a scalar Gaussian noise with zero mean and variance $\sigma_n^2 = 0.2^2$.

6.1 Scenario Description

In the considered scenario, $N = 12$ acoustic sensors are located as in Table 1. A single target is moving on the road and $K = 30$ measurements are collected

Table 1: *Sensor positions*

n	1	2	3	4	5	6
$s_x^n\ [m]$	77.0	88.6	43.3	59.9	10.0	30.6
$s_y^n\ [m]$	7.0	22.2	10.0	16.3	10.2	16.8
n	7	8	9	10	11	12
$s_x^n\ [m]$	23.4	9.1	46.0	37.7	78.1	65.5
$s_y^n\ [m]$	34.9	40.0	41.4	50.3	52.3	36.7

for every n-th sensor. The measurements are related to the target positions x_k at time steps $k = 1 : K$ with units in seconds. The scenario is presented in Figure 3. Acoustic power measurements are obtained by first taking the square of the sound signal from each sensor and then averaging it for each $T = 1$ second. As an example, the raw sound data and the acoustic power measurements generated from it, for sensor nr 6, are presented in Figure 4.

The ground truth reference target state is obtained using GPS.

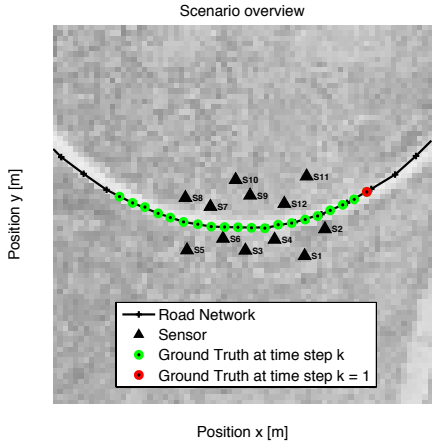


Figure 3: *Target trajectory and sensor positions.*

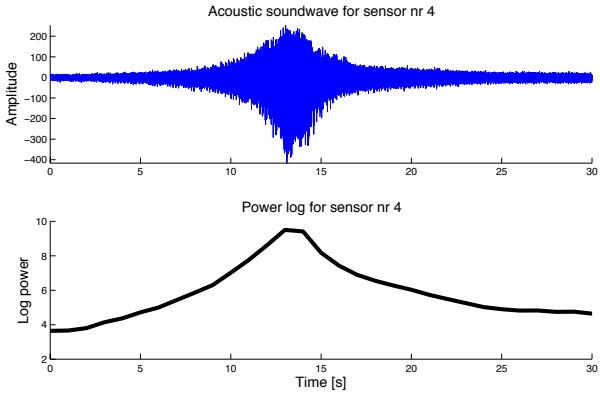


Figure 4: *Acoustic soundwave and logarithmic power measurements.*

6.2 Results

In the experiment, position biases are added to the sensors by simply switching the positions of two pairs of sensors (3–4 and 7–8, respectively). Also the prior knowledge about biases is introduced for each n-th sensor as zero mean $\theta_\pi^n = [\theta_{x,\pi}^n \quad \theta_{y,\pi}^n \quad \theta_{g,\pi}^n]^T = [0 \quad 0 \quad 0]^T$ with covariance $P_\pi^n = \mathrm{diag}([5^2 \quad 5^2 \quad 0.1^2])$. A total number of $Mc = 70$ Monte Carlo runs of the particle filter are performed for each case, with $I = 30$ number of EM iterations. The number of particles in each case is equal to $M = 500$. The results are described below.

The RMS error based on Monte Carlo runs is presented in Figure 5. Numerical results for selected biases are presented in Table 2. Figure 6 presents true and predicted logarithmic power measurements for sensor 4 before and after calibra-

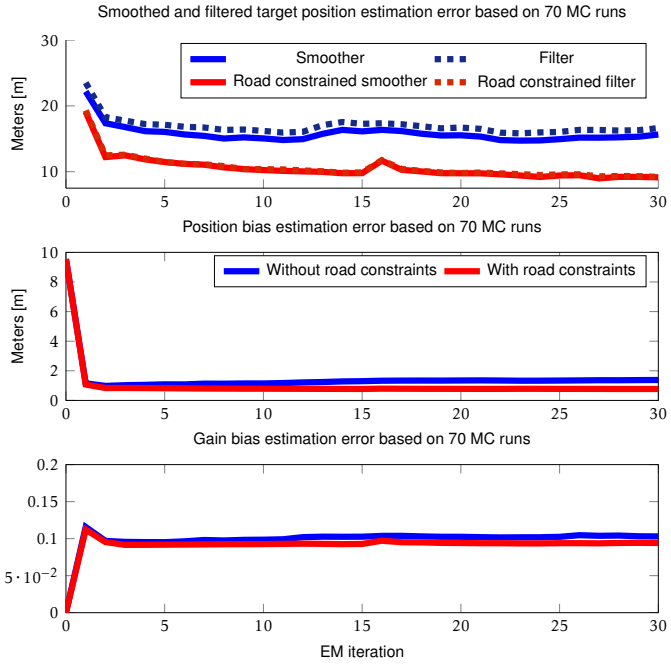


Figure 5: *Mean RMS error based on Monte Carlo runs.*

Table 2: *Monte Carlo results for bias estimates in simulated scenario*

Bias index	True bias	EM algorithm Without constraints	EM algorithm With constraints
θ_x^3	16.67	12.64 ± 3.24	11.03 ± 2.37
θ_y^3	6.25	1.48 ± 4.23	4.13 ± 0.89
θ_g^3	0	0.01 ± 0.04	-0.04 ± 0.01
θ_x^4	-16.67	-15.3 ± 3.79	-16.42 ± 3.49
θ_y^4	-6.25	-3.53 ± 3.7	-6.16 ± 1.29
θ_g^4	0	0.04 ± 0.04	0.02 ± 0.01
θ_x^7	-14.27	-10.26 ± 3.22	-11.44 ± 2.43
θ_y^7	5.08	6.88 ± 6.09	4.59 ± 0.99
θ_g^7	0	-0.01 ± 0.03	0.01 ± 0.01
θ_x^8	14.27	16.09 ± 4.21	16.15 ± 3.4
θ_y^8	-5.08	-9.74 ± 5.34	-10.11 ± 2.15
θ_g^8	0	-0.16 ± 0.04	-0.15 ± 0.03

tion for both unconstrained and road constrained version of the EM algorithm for one of the MC runs. Figure 7 presents filtered and smoothed estimates for $I = 30^{th}$ iteration of the EM algorithm (after calibration) compared to ground truth.

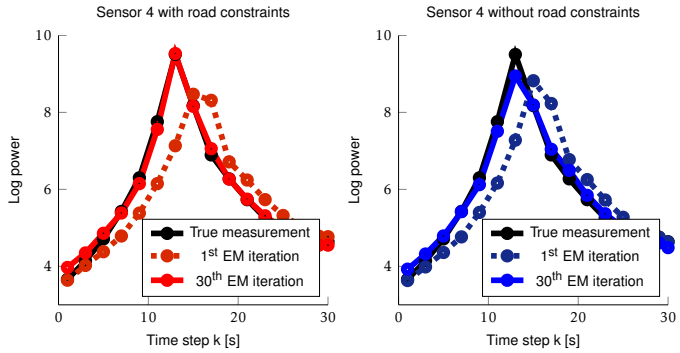


Figure 6: *Predicted power measurements (obtained from smoothed state estimates) before and after calibration.*

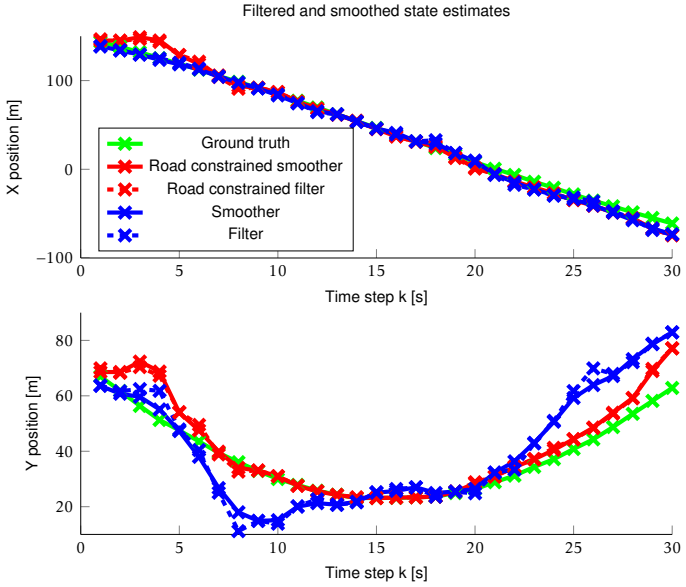


Figure 7: *Filtered and smoothed estimates after calibration together with the ground truth.*

7 Conclusions

In this paper, the EM algorithm was presented as a general solution to calibrate ground sensor network without special equipment, both natural and opportunistic targets can be used. Application of road constraints to the particle filter provides improvement in bias estimation quality, especially in terms of the standard deviation of the bias estimates. It provides also much better state estimates for the target in the scenario with missed detections.

Acknowledgment

The research leading to these results has received funding from the EU's Seventh Framework Program under grant agreement no 238710. The research has been carried out in the MC IMPULSE project: https://mcimpulse.isy.liu.se/.

Bibliography

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

S Dhar. Application of a recursive method for registration error correction in tracking with multiple sensors. In *American Control Conference, 1993*, pages 869–874. IEEE, 1993.

Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.

Fredrik Gustafsson, Umut Orguner, Thomas B Schön, Per Skoglar, and Rickard Karlsson. Navigation and tracking of road-bound vehicles. *Handbook of Intelligent Vehicles*, 2012.

M Ignagni. An alternate derivation and extension of Friendland's two-stage Kalman estimator. *Automatic Control, IEEE Transactions on*, 26(3):746–750, 1981.

Zhenhua Li, H Leung, and Liang Tao. Simultaneous registration and fusion of radar and ESM by EM-EKS. In *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, pages 1130–1134. IEEE, 2010.

Branko Ristic, Sanjeev Arulampalm, and Neil James Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.

David Salmond, Martin Clark, Richard Vinter, and Simon Godsill. Ground target modeling, tracking and prediction with road networks. In *Information Fusion, 2007 10th International Conference on*, pages 1–8. IEEE, 2007.

Thomas B Schön, Adrian Wills, and Brett Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011.

Egils Sviestins. Online bias estimation for multisensor tracking. In *Information, Decision and Control, 1999. IDC 99. Proceedings. 1999*, pages 221–226. IEEE, 1999.

M. Syldatk, E. Sviestins, and F. Gustafsson. Expectation maximization algorithm for calibration of ground sensor networks using a road constrained particle filter. In *Proceedings of the 15th International Conference on Information Fusion (FUSION)*, pages 771–778, Singapore, July 2012.

Paper B

Simultaneous Tracking and Sparse Calibration in Ground Sensor Networks using Evidence Approximation

Authors: Marek Syldatk, Fredrik Gustafsson

Edited version of the paper:

M. Syldatk and F. Gustafsson. Simultaneous tracking and sparse calibration in ground sensor networks using evidence approximation. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

Simultaneous Tracking and Sparse Calibration in Ground Sensor Networks using Evidence Approximation

Marek Syldatk, Fredrik Gustafsson

Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
marek.syldatk@liu.se
fredrik.gustafsson@liu.se

Abstract

Calibration of ground sensor networks is a complex task in practice. To tackle the problem, we propose an approach based on simultaneous tracking of targets of opportunity and sparse estimation of the bias parameters. The evidence approximation method is used to get a sparse estimate of the bias parameters, and the method is here extended with a novel marginalization step where a state smoother is invoked. A simulation study shows that the non-zero bias parameters are detected and well estimated using only one target of opportunity passing by the network.

1 Introduction

Calibration of the ground sensor network (GSN), also known as sensor registration or bias estimation, is a crucial element for performance of the entire system. Improper alignment of the sensors might decrease the performance of the network and in fact result in degrading the quality of tracking, appearance of ghost tracks and problems in measurement to track association.

There are many previous works dealing with bias estimation (Dhar (1993), Sviestins (1999), Okello and Ristic (2003), Vermaak et al. (2005)). Here we focus on methods applicable to GSN. The different biases, here denoted bias parameters, may include location errors, orientation errors, range measurements etc., depending on sensor type. There are two main groups of techniques for calibrating the sensors: using reference targets, or using targets of opportunity. Both can be performed on-line or off-line.

In this application, the reference targets could typically be one or more vehicles carrying satellite navigation equipment, e.g. GPS. One must make sure that

enough measurements are generated for each sensor with the target at different locations. Once the data has been collected, one can apply e.g. a maximum likelihood (ML) method (Okello and Ristic, 2003), to estimate the bias parameters that make observations of the target fit the reference data best.

In case no reference targets can be used, one needs to track targets of opportunity for the purpose of bias estimation. One approach is to simultaneously estimate the track states and biases, which can be accomplished by forming augmented state vectors that combine target state estimates together with bias estimates (Dhar, 1993). With a large number of targets and sensors this will hardly be tractable due to computational requirements etc. There are however suboptimal but very efficient techniques to decouple target state and bias estimation process, fully feasible for on-line operation (Ignagni (1981), Sviestins (1999), Vermaak et al. (2005)).

As the number of measurements suitable for bias estimation can be quite low, it is desirable to find a method that uses them as effectively as possible. One such off-line method is the expectation maximization (EM) algorithm (Li et al. (2010), Schön et al. (2011), Syldatk et al. (2012)), where the entire set of measurements is processed iteratively to provide both state estimates and bias parameters.

This paper examines a slightly different approach to the off-line bias estimation with targets of opportunity. Because the number of measurements is usually low, it is useful to apply some regularization to the maximum likelihood estimation, in order to avoid the problems with observability. In practice, only a few sensors need calibration usually, and most of the bias parameters are zero, so one should use this extra information.

In this paper biases are considered to be stochastic variables and a method called *Type-II Bayes* (Berger, 1985), *evidence approximation* (MacKay, 1992) or *sparse Bayesian learning* (Wipf and Rao, 2004) is applied, where each bias parameter has its own regularization parameter, corresponding to the priors in a Bayesian framework. Those parameters are estimated together with bias parameters. The approach utilizes *Occam's razor* (MacKay (1992), Bishop (2006)), which lets us find a good balance between model complexity and fit to data. It also implies sparsity through regularization and, in addition, provides us a very useful information about how well each of the parameters is determined by the data.

Application of different regularizers for each parameter is a basic idea underlying the Relevance Vector Machines algorithm, provided in Tipping (2001), using the same sparse Bayesian learning framework by MacKay (1992).

Our approach to simultaneous tracking and calibration requires a novel extension to the evidence approximation method, where the target trajectories are marginalized using state smoothers for computing the evidence function.

The method will be described in detail in Section 3 after the formal definition in Section 2. Section 4 provides simulation results. In Section 5 final conclusions are stated and future work is suggested.

2 Problem Formulation

A ground target is detected by a number of sensors, resulting in a set of observations $Y_K = \{y_k\}_{k=1}^K$ of the target states $X_K = \{x_k\}_{k=1}^K$, where y_k is a single measurement, or set of measurements stacked as a vector, and x_k is target state at time $k = 1 \dots K$. The measurements are affected by a number of bias parameters and by a measurement noise. All the bias parameters, for all sensors, are collected in a single vector θ.

The measurement model for each sensor is assumed to be a known, nonlinear function $h(x_k, \theta)$ of the state x_k, with measurement bias vector stated as $\theta = [\theta_1 \dots \theta_m \dots \theta_M]^T$, and is defined as

$$y_k = h(x_k, \theta) + \nu_k, \tag{1}$$

where ν_k is an additive noise with Normal distribution and known covariance matrix R.

As was partially mentioned in the introduction, there are three main problems with calibration of the sensor network. Small number of measurements causes problems with observability, and thus the maximum likelihood methods do not usually provide reliable results. The problem can be solved by applying some constraints on the estimates, which in the Bayesian framework are solved by assigning the priors. Choosing a correct prior is also a problem by itself, since it should utilize the sparsity feature of the bias vector. Finally, a correct calibration cannot be performed, when the state of the ground target is unknown, which is usually the case.

2.1 General Estimation Framework

In the estimation approach, the bias vector θ is assumed to be a stochastic variable. The method consists of finding a posterior distribution of the bias parameters, by using the Bayesian maximum a posteriori method

$$p(\theta|Y_K, \alpha) = \frac{p(Y_K|\theta, \alpha)p(\theta|\alpha)}{p(Y_K|\alpha)}. \tag{2}$$

The set of *hyper-parameters* α, determines the prior distribution for θ, given by $p(\theta|\alpha)$. Since those parameters are in general unknown, a method using maximization of an *evidence* function (also called a *marginal likelihood*) is applied. Evidence, which is a normalizing term in (2), is obtained by integrating out the parameter vector, as

$$p(Y_K|\alpha) = \int p(Y_K|\theta, \alpha)p(\theta|\alpha)d\theta. \tag{3}$$

A general framework for the procedure of maximizing the evidence, called *evidence procedure*, is provided for example in MacKay (1992), Tipping (2001) or Gull and Skilling (1999).

In this paper a general case is handled, when the likelihood $p(Y_K|\theta,\alpha)$ depends also on the state of the target X_K, that is generally unknown. The set of those target states is then integrated out from $p(Y_K|\theta,\alpha,X_K)$ by making use of an estimated state distribution $\hat{p}(X_K)$, obtained from a filtering/smoothing algorithm (Ristic et al. (2004), Mahler (2007), Gustafsson (2012)). Thus the likelihood, independent of X_K, is computed using

$$p(Y_K|\theta,\alpha) = \int p(Y_K|X_K,\theta,\alpha)\hat{p}(X_K)dX_K. \tag{4}$$

3 Sparse Bayesian Calibration Algorithm

Having previously defined the measurements in (1) and using the assumption that measurements are independent, the likelihood for full data set can be easily written as

$$p(Y_K|X_K,\theta,\alpha) = \mathcal{N}(\mathbf{Y};\mathbf{h}(\mathbf{X},\theta),\mathbf{R}), \tag{5}$$

where $\mathbf{Y} = [y_1^T \dots y_K^T]^T$, $\mathbf{X} = [x_1^T \dots x_K^T]^T$, $\mathbf{h}(\mathbf{X},\theta) = [h(x_1,\theta)^T \dots h(x_k,\theta)^T]^T$ and $\mathbf{R}$ is a covariance matrix with R matrices on diagonal.

In case of the sensor networks, the true value of the state x_k is usually unknown, so it is reasonable to use estimates obtained from for instance the Kalman or particle filter/smoother, using some initial value of θ^0 and α^0. For the Gaussian case, a set of state estimates $\hat{X}_K = \{\hat{x}_k\}_{k=1}^K$, together with its corresponding covariance matrices $\hat{P}_K = \{\hat{P}_k\}_{k=1}^K$, results in a distribution

$$P(X_K|Y_K,\theta^0,\alpha^0) = \mathcal{N}(\mathbf{X}|\hat{\mathbf{X}},\hat{\mathbf{P}}), \tag{6}$$

where $\hat{\mathbf{X}} = [\hat{x}_1^T \dots \hat{x}_K^T]^T$ and $\hat{\mathbf{P}}$ is block diagonal with $\hat{P}_1^T \dots \hat{P}_K^T$ on the diagonal. By treating (6) as a distribution over the state, $\hat{p}(X_K) = P(X_K|Y_K,\theta^0,\alpha^0)$, one can remove the dependence on X_K from (5), by applying the marginalization defined in (4)

By linearizing the measurement function around some $\hat{\theta}$ and $\hat{\mathbf{X}}$, Equation (1) can be rewritten, using a 1st order Taylor expansion, as

$$\bar{y}_k = H_\theta(\hat{x}_k,\hat{\theta})\theta + H_x(\hat{x}_k,\hat{\theta})(x-\hat{x}_k) + \nu_k, \tag{7}$$

where $\bar{y}_k = y_k - h(\hat{x}_k,\hat{\theta}) + H_{\hat{\theta}}(\hat{x}_k,\hat{\theta})\hat{\theta}$ and

$$H_\theta(\hat{x}_k,\hat{\theta}) = \left.\frac{\partial h(x_k,\theta)}{\partial\theta}\right|_{\theta=\hat{\theta},x=\hat{x}_k}, \tag{8}$$

$$H_x(\hat{x}_k,\hat{\theta}) = \left.\frac{\partial h(x_k,\theta)}{\partial x_k}\right|_{\theta=\hat{\theta},x=\hat{x}_k}. \tag{9}$$

Thus the result of integration, where some terms got cancelled in the meantime, is given by

$$p(Y_K|\theta,\alpha) \approx \mathcal{N}(\bar{\mathbf{Y}}|\mathbf{H}_\theta\theta,\bar{\mathbf{R}}), \tag{10}$$

where the covariance matrix $\bar{\mathbf{R}}$ is defined as $\bar{\mathbf{R}} = (\mathbf{R} + \mathbf{H_x}\hat{\mathbf{P}}\mathbf{H_x^T})$, and $\bar{\mathbf{Y}} = [\bar{y}_1 \dots \bar{y}_K]^T$, $\mathbf{H}_\theta = [H_\theta(\hat{x}_1, \hat{\theta})^T \dots H_\theta(\hat{x}_K, \hat{\theta})^T]^T$ and $\mathbf{H_X}$ is defined analogously.

In a Bayesian framework, to infer the values of θ, we need to apply some constraint on the data, which is accomplished by assigning a prior. Here we use a zero mean Gaussian prior

$$p(\theta|\alpha) = \prod_{m=1}^{M} \mathcal{N}(\theta_m|0, \alpha_m) = \mathcal{N}(\theta|0, A^{-1}), \tag{11}$$

where $\alpha = [\alpha_1 \dots \alpha_M]^T$ is the hyper-parameter defining the precision, and $A = \text{diag}(\alpha_1, \dots, \alpha_M)$. The problem is how to assign the hyper-parameters? In this framework, the hyper-parameters are assumed unknown and inferred from the data together with the set of parameters θ. Therefore, by having defined the likelihood and prior above, one only needs to define the prior for hyper-parameters. Here we will consider flat prior that is non informative (Jeffreys, 1946), $p(\alpha) = const$, giving equal probability for all possible values of α. Having defined the prior, we can now proceed to the Bayesian inference of unknown variables.

3.1 Parameter Estimation

In this step the values of θ will be inferred from the data. In a fully Bayesian framework, the posterior over the parameter set θ should be obtained by integrating out the hyper-parameters, so then

$$p(\theta|Y_K) = \int p(\theta|Y_K, \alpha) p(\alpha|Y_K) d\alpha. \tag{12}$$

By assuming the posterior for α is sharply peaked around its estimate $\hat{\alpha}$, so $p(\alpha) \approx \delta(\alpha - \hat{\alpha})$, we can (MacKay, 1992) use the approximation

$$p(\theta|Y_K) \simeq p(\theta|Y_K, \alpha)|_{\alpha=\hat{\alpha}}. \tag{13}$$

Hereby the posterior above can be rewritten, using Bayes rule, as

$$p(\theta|Y_K) \simeq \left. \frac{p(Y_K|\theta, \alpha) p(\theta|\alpha)}{p(Y_K|\alpha)} \right|_{\alpha = \hat{\alpha}}. \tag{14}$$

The posterior for θ is obtained using (10) and (11), together with a simple rule for the posterior distribution in a Gaussian case, as

$$p(\theta|Y_K) = \mathcal{N}(\theta|\mu, \Sigma), \tag{15}$$

which is also Gaussian, with mean and covariance defined by

$$\mu = \Sigma(\mathbf{H}_\theta^\mathbf{T} \bar{\mathbf{R}}^{-1} \bar{\mathbf{Y}}), \tag{16}$$

$$\Sigma = (\mathbf{H}_\theta^\mathbf{T} \bar{\mathbf{R}}^{-1} \mathbf{H}_\theta + A)^{-1}. \tag{17}$$

It is sometimes useful to obtain a point estimate of the bias vector, which maximizes the posterior

$$\hat{\theta}_{map} = \arg\max_{\theta} p(\theta|Y_K), \tag{18}$$

which is equal to the mean defined in (16).

3.2 Hyper-parameter Estimation

To infer the hyper-parameters, Bayes rule is applied again, so

$$p(\alpha|Y_K) = \frac{p(Y_K|\alpha)p(\alpha)}{p(Y_K)}. \tag{19}$$

As was declared before, the prior $p(\alpha)$ is assumed to be flat, so to obtain optimal values of hyper-parameters, we only need to consider the first term in a nominator, $p(Y_K|\alpha)$. As one can easily notice, it is the normalizing constant of Equation (14), defined previously in (3) as an *evidence*. To obtain the evidence, the parameter θ needs to be integrated out as in (3), so using (10) and (11), one can write

$$p(Y_K|\alpha) = \int \mathcal{N}(\bar{\mathbf{Y}}|\mathbf{H}_\theta\theta, \bar{\mathbf{R}})\mathcal{N}(\theta|0, A^{-1})d\theta \tag{20}$$

and the optimal value is then the one that maximizes the likelihood

$$\hat{\alpha} = \arg\max_{\alpha} p(Y_K|\alpha). \tag{21}$$

Because (20) is a simple convolution of Gaussians, the evidence can be easily computed as

$$p(Y_K|\alpha) = \frac{|\bar{\mathbf{R}}|^{-\frac{1}{2}}}{(2\pi)^{\frac{Kd_y}{2}}}|A|^{\frac{1}{2}}|\Sigma|^{\frac{1}{2}}e^{-Q(\bar{\mathbf{Y}})}, \tag{22}$$

where d_y is a dimension of $\bar{\mathbf{Y}}$, and the quadratic term $Q(\bar{\mathbf{Y}})$ is given by

$$Q(\bar{\mathbf{Y}}) = \frac{1}{2}(\bar{\mathbf{Y}}^{\mathbf{T}}\bar{\mathbf{R}}^{-\mathbf{1}}\bar{\mathbf{Y}} - \mu^T\Sigma^{-1}\mu). \tag{23}$$

Now, to obtain the estimate of α, the approach presented in MacKay (1992) will be used. By taking the derivative of a logarithm of the evidence (22) with respect to α_m, and equalling it to zero, we get

$$-\frac{1}{2}\mu_m^2 - \frac{1}{2}\Sigma_{mm} + \frac{1}{2\alpha_m} = 0, \tag{24}$$

where μ_m is the m-th element of the m vector, defined in (16), and Σ_{mm} is the m-th diagonal element of the covariance matrix defined in (17). The equation yields a following solution

$$\alpha_m = \frac{\gamma_m}{\mu_m^2}, \tag{25}$$

where $\gamma_m = 1 - \alpha_m\Sigma_{mm}$.

According to MacKay (1992) or Bishop (2006), the new parameter can take values in a range of $\gamma_m \in [0 \dots 1]$, and it determines how well the corresponding parameter θ_m is determined by the data. For small standard deviation of the prior, when α_m is large, the estimates are strongly constrained by the prior. In that case Σ_{mm} in (17) is dominated by hyper-parameter, so $\Sigma_{mm} \approx \alpha_m^{-1}$, and thus $\gamma_{mm} \approx 0$. On the other hand, when α_m takes a small value, which means the corresponding estimate μ_m is well fit to the data, then $\gamma_m \approx 1$.

As we can see the algorithm requires an iterative approach. We need to start with some initial estimates of θ and α and iteratively refine the estimates. The procedure is shown in Algorithm 1.

Algorithm 1 Sparse calibration algorithm

1. Initiate at $i = 0$ with θ^0 and α^0.
2. Use a filtering/smoothing algorithm to obtain state estimates $\hat{X}_K$ and corresponding covariances $\hat{P}_K$, using θ^0 as a bias estimate.
3. For $i = 1$:
   (a) Linearize the measurement function as in (7) using $\hat{X}_K$ and $\hat{\theta}^{i-1}$.
   (b) Set $\alpha = \hat{\alpha}^{i-1}$ and using Equation (16), (17) and (18) obtain new estimate of bias vector $\hat{\theta}^i = \hat{\theta}_{map} = \mu$.
   (c) Set $\gamma_m = 1 - \hat{\alpha}_m^{i-1}\Sigma_{mm}$ and obtain new estimate of α using (25), given by $\hat{\alpha}_m^i = \gamma_m/\mu_m^2$
   (d) If converged: (4); Otherwise: $i = i + 1$ and repeat (3).
4. End iterations.

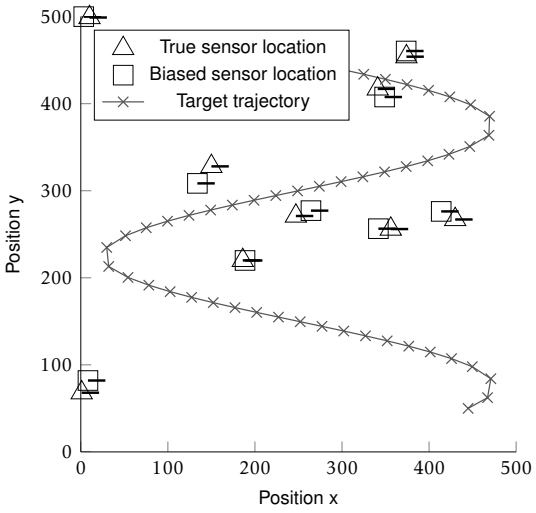


Figure 1: *Scenario overview with target ground truth trajectory together with true and biased sensor positions.*

4 Experimental Results

In the experimental scenario, $N_S = 10$ sensors are distributed randomly on an area of 500×500 meters. Sensors are gathering range measurements (in meters) and bearing measurements (in radians). Both positions and measurements are biased, with additive biases. Positioning biases related to the x- and y-axis are defined as θ_x and θ_y respectively; measurement biases related to range- and bearing-bias are defined as θ_r and θ_a respectively. Simulations are performed in a Monte Carlo (MC) manner, with biases randomly generated in each MC run, according to zero-mean normal distribution with standard deviations σ_x, σ_y, σ_r and σ_a respectively. Bias vector is also assumed to be sparse, with sparsity s_f, representing a fraction of non-zero elements, ranging from 0 to 1. Measurements, as in (1), are collected using a following vector measurement function, defined for each sensor as

$$h(x_k) = \begin{bmatrix} h_1(x_k) \\ h_2(x_k) \end{bmatrix} = \begin{bmatrix} \sqrt{d_{x,k}^2 + d_{y,k}^2} + \theta_r \\ \arctan \frac{d_{y,k}}{d_{x,k}} + \theta_a \end{bmatrix}, \tag{26}$$

$$d_{x,k} = p_{x,k}^x - (p_x^s - \theta_x), \quad d_{y,k} = p_{y,k}^x - (p_y^s - \theta_y), \tag{27}$$

and where state vector $x_k = [p_{x,k}^x, p_{y,k}^x, v_{x,k}^x, v_{y,k}^x]^T$, with elements corresponding to x- and y- position and x- and y- velocities of the target at time step k respectively and p_x^s and p_y^s are positions of the sensor in x- and y- axis. An overview

Table 1: *Parameters of the scenario.*

Parameter	Symbol	Value
Number of measurements per sensor	K	50
Number of sensors	N_S	10
Range measurement noise std	σ_y^r	10 m
Angle measurement noise std	σ_y^r	10°
Number of MC runs	MCn	250
Sparsity range	s_p	0 : 0.05 : 1
Number of iterations	I	25
EKF initial position std	σ_p	10 m
EKF initial velocity std	σ_v	5 m/s
Bias: x/y position std	σ_x / σ_y	10 m / 10 m
Bias: range/angle std	σ_r / σ_a	10 m / 10°

of the simulated scenario, with $K = 50$ of measurements, is presented in Figure 1. Parameters of the scenario are summarized in Table 1. For parameter estimation, smoothed state estimates $\hat{X}_K$ are obtained from Kalman smoother using a standard constant velocity (CV) model.

Figure 2 presents the summary RMS error (mean + 1 standard deviation) of bias estimation results, where maximum likelihood (ML), maximum a posteriori (MAP) and evidence approximation (EA) methods are compared for different bias vector sparsities. In the case of MAP, the true prior, from which biases were generated,

was used. Figure 3 presents detailed results for one Monte Carlo simulation.

4.1 Summary

As we can see in Figure 2, the proposed method provides much better results compared to traditional maximum likelihood, and slightly, but consistently, performs worse than the MAP method. In the simulation, MAP is assumed to use the true prior distribution from which the biases were generated, including the knowledge about the sparsity shape of the bias vector. It is important to notice, that the more the bias vector contains zero element ($s_p \to 0$), the better the EA and MAP methods perform compared to ML. It is due to the fact, that both methods utilize the feature of sparsity of the bias vector, which is implied by using the correct prior (MAP) or through Occam's razor (EA).

On the other hand, ML does not utilize this feature, and since there are no priors applied, the method tends to provide unreliable results in case, when the estimated parameters are not well determined by the measurement data.

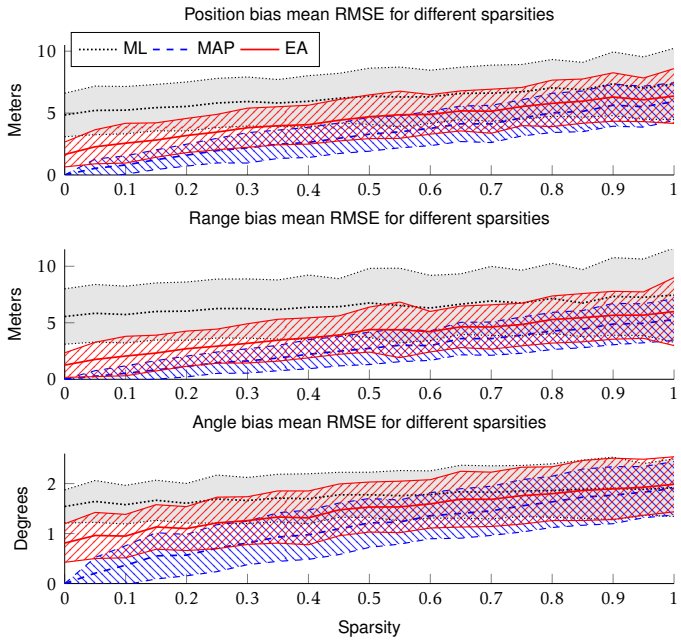


Figure 2: *Mean and standard deviation of the bias estimation RMS Error for different sparsities.*

Figure 3 presents one Monte Carlo run estimation results for sensors 5, 6 and 7, with corresponding bias parameters θ_{17} to θ_{28} and related γ_m values obtained from the EA algorithm. In that case only parameters number 18, 21, 26 and 27 had non-zero values. As we can see, for most θ_m all 3 algorithms managed to

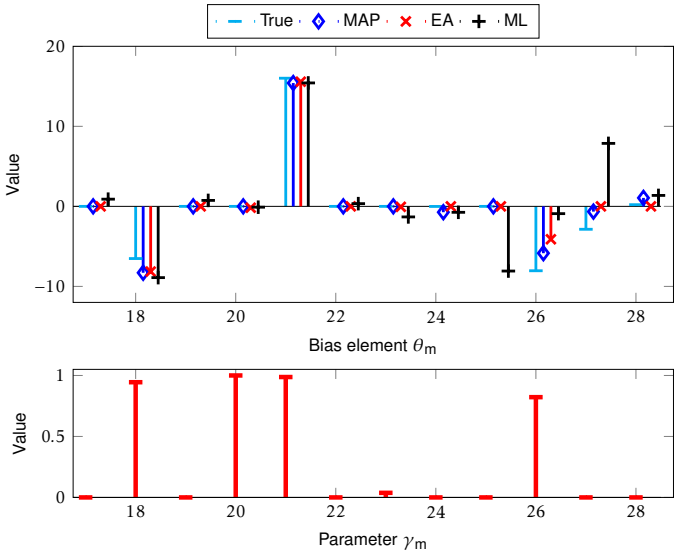


Figure 3: *Comparison of estimation methods and γ parameter.*

provide quite similar results. We shall now look at the difference that appeared between them.

One can observe that EA algorithm correctly detected most of the zero and non-zero elements. The parameters, that were well determined by the measurements, are indicated by the value of a corresponding gamma parameter close to unity. One can observe this situation for θ_{21}, where all methods give very good results, and the parameter is well determined by the data, so $\gamma_{21} \simeq 1$. On the other hand, for elements θ_{25} and θ_{27}, the ML has significantly diverged, most probably due to the problem with observability. In that case EA algorithm performed more robust, and has shrunk the values towards zero, what is indicated by value of $\gamma_{25,27} \simeq 0$. The parameters that were estimated with values close to zero, unless they were well determined by the data, were shrunk towards the prior mean value (zero), and therefore the sparsity feature was implied. Situation like this can be observed for example for parameters $\theta_{22,23,24}$.

5 Conclusions

The proposed evidence approximation (EA) method estimates the bias parameters in a sensor calibration problem, jointly with the state trajectories of targets of opportunity. As a bonus, EA estimates the prior of the bias parameters, which can in itself serve as an indicator of how well the bias parameters are estimated. This is a useful and practical advantage compared to the more direct maximum likelihood (ML) approach. Further, the EA method provides a sparse bias vector in contrast to the ML method, which is useful in practical ground sensor network deployments, where many sensors can be assumed bias-free.

Bibliography

James O Berger. *Statistical decision theory and Bayesian analysis*. Springer, 1985.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

S Dhar. Application of a recursive method for registration error correction in tracking with multiple sensors. In *American Control Conference, 1993*, pages 869–874. IEEE, 1993.

SF Gull and J Skilling. Quantified Maximum Entropy MemSys5 Users' Manual. *Maximum Entropy Data Consultants Ltd., Suffolk*, 1999.

F. Gustafsson. *Statistical Sensor Fusion*. Utbildningshuset/Studentlitteratur, 2012. ISBN 9789144077321.

M Ignagni. An alternate derivation and extension of Friendland's two-stage Kalman estimator. *Automatic Control, IEEE Transactions on*, 26(3):746–750, 1981.

Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

Zhenhua Li, H Leung, and Liang Tao. Simultaneous registration and fusion of radar and ESM by EM-EKS. In *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, pages 1130–1134. IEEE, 2010.

David JC MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

Ronald PS Mahler. *Statistical multisource-multitarget information fusion*, volume 685. Artech House Boston, 2007.

Nickens Okello and Branko Ristic. Maximum likelihood registration for multiple dissimilar sensors. *Aerospace and Electronic Systems, IEEE Transactions on*, 39 (3):1074–1083, 2003.

Branko Ristic, Sanjeev Arulampalm, and Neil James Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.

Thomas B Schön, Adrian Wills, and Brett Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011.

Egils Sviestins. Online bias estimation for multisensor tracking. In *Information, Decision and Control, 1999. IDC 99. Proceedings. 1999*, pages 221–226. IEEE, 1999.

M. Syldatk and F. Gustafsson. Simultaneous tracking and sparse calibration in ground sensor networks using evidence approximation. In *Proceedings of the*

38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vancouver, Canada, May 2013.

M. Syldatk, E. Sviestins, and F. Gustafsson. Expectation maximization algorithm for calibration of ground sensor networks using a road constrained particle filter. In *Proceedings of the 15th International Conference on Information Fusion (FUSION)*, pages 771–778, Singapore, July 2012.

Michael E Tipping. Sparse Bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

Jaco Vermaak, Simon Maskell, and Mark Briers. Online sensor registration. In *Aerospace Conference, 2005 IEEE*, pages 2117–2125. IEEE, 2005.

David P Wipf and Bhaskar D Rao. Sparse Bayesian learning for basis selection. *Signal Processing, IEEE Transactions on*, 52(8):2153–2164, 2004.

Paper C

Expectation Maximization Algorithm for Simultaneous Tracking and Sparse Calibration of Sensor Networks

Authors: Marek Syldatk, Fredrik Gustafsson

Edited version of the paper:

M. Syldatk and F. Gustafsson. Expectation maximization algorithm for simultaneous tracking and sparse calibration of sensor networks. In *IEEE Transactions on Signal Processing (TSP)*. To be submitted.

Expectation Maximization Algorithm for Simultaneous Tracking and Sparse Calibration of Sensor Networks

Marek Syldatk, Fredrik Gustafsson

Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
marek.syldatk@liu.se
fredrik.gustafsson@liu.se

Abstract

The performance of ground sensor networks (GSN) relies on accurate knowledge of sensor positions and sensor linearity parameters. Calibration of these should be done automatically after the deployment and later on regular intervals. Preferably, only targets of opportunity should be used to avoid manual intervention. We present a Bayesian solution that scales well with network size and efficiently utilizes a limited number of data to get a sparse estimate of a potentially huge number of calibration parameters. The approach is based on iteratively solving two different estimation problems using the expectation maximization (EM) algorithm. First, the target trajectory and sensor parameters are estimated for a given prior on the parameters. Second, the prior is estimated using the evidence approximation (EA) principle to get a sparse solution. The algorithm successfully jointly estimates the sparsity structure, calibration parameters and target trajectory as demonstrated on both simulated and real data from a microphone network with a passing vehicle.

1 Introduction

Ground sensor networks (GSN) are a versatile solution to many complex surveillance problems. The network can consist of microphones, radars, seismometers, radio transmitters/receivers *etc.* In all these cases, the position and linearity parameters such as offset and scale are important to know for the target tracking and localization algorithms needed in the surveillance applications. For the radar and sensor arrays also the orientation is crucial. This will be referred to as the calibration problem, which is sometimes also called bias estimation or sensor registration.

The problem of sensor calibration is widely considered in the literature (Dhar (1993), Sviestins (1999), Okello and Ristic (2003), Vermaak et al. (2005)). There exist multiple approaches to the problem, including on-line and off-line methods. Both types can be further divided into methods using reference targets (for example beacons with known positions or a target with GPS receiver) or targets of opportunity (where the target position is unknown). In the simplest case, when the reference data is available and measurements are collected by all the sensors, one can simply apply maximum likelihood (ML) method (or maximum a posteriori (MAP) if prior information is available) in order to obtain estimates of bias parameters.

On the other hand, when a reference target cannot be used, in order to perform bias estimation, one needs to simultaneously estimate both target states and calibration parameters. It can be achieved for example by augmenting the target state vector with the parameters we want to estimate and then applying a state estimation filter (Dhar, 1993). This approach though might cause some problems, especially in the case of a large number of targets and unknown parameters. However, there exist some suboptimal, and efficient methods to decouple those two problems, fully feasible for online application (Ignagni (1981), Sviestins (1999), Vermaak et al. (2005)).

Another approach is to use an expectation maximization algorithm (Dempster et al. (1977), Li et al. (2010), Schön et al. (2011), Kung et al. (2005)), that is an offline method designed to solve the maximum likelihood (or maximum a posteriori) estimation problem in the presence of latent variables. In application to sensor registration, the method estimates the state of the target (the latent variable) and provides ML estimates of the parameter in an iterative manner, with guaranteed convergence to (at least) local minimum.

In practical applications, the usual number of measurements is relatively low, and thus problems with observability of bias parameters might appear. It is thus natural to apply some regularization to the maximum likelihood estimation, that in Bayesian framework is evaluated through priors. In real life applications it is also common, that most of the sensors are already correctly calibrated (biases are close to zero) and one should also incorporate this information to improve estimation quality.

This paper considers quite a complex situation, when both prior information and reference targets are unavailable. The bias parameters are considered to be stochastic variables, and a method called *Type-II Bayes* (Berger, 1985), *evidence approximation* (MacKay, 1992) or *sparse Bayesian learning* (Wipf and Rao, 2004) is applied, where each bias parameter has its own regularization parameter, corresponding to the application of priors in the Bayesian framework. Those regularization parameters, also called hyper-parameters, are estimated together with the parameters. Because of presence of latent variables, the EM algorithm is applied to tackle that situation. A crucial feature of the proposed algorithm is that it utilizes *Occam's razor*, which lets us find a good balance between model complexity and fit to data and which implies the sparsity through regularization.

Application of different regularizers for every parameter is a core idea underlying the Relevance Vector Machines algorithm by Tipping (2001), that uses the same sparse Bayesian learning framework presented in MacKay (1992).

The formal definition of the problem is presented in Section 2, together with a detailed description of a method in Section 3. Section 4 provides simulation results supported by real world experiment results in Section 5. Final conclusions are stated in Section 6 together with suggestions for future work.

2 Problem Formulation

2.1 Notation and Estimation Framework

A target is detected by a number of sensors, resulting in a set of observations $Y_K = \{y_k\}_{k=1}^K$ corresponding to the set of target states $X_K = \{x_k\}_{k=1}^K$, where y_k is a single measurement or set of measurements stacked as a vector, and x_k is a state of the target at time instant k. Measurements are affected both by a measurement noise and a set of (unknown) bias parameters. All the biases are collected in a single bias parameter vector θ that is assumed to be at least partially sparse, which means that a number of elements are equal to zero. It corresponds to a situation, when only part of sensors within the network require calibration. One Bayesian approach to model sparsity is via a prior variance α_m for each parameter θ_m.

In summary, we have the unknown target trajectory X_K, bias parameters θ and prior parameter precisions α (by precision we understand an inverse of a variance). There are now several possible approaches:

- One can solve the un-regularized problem using the Expectation Maximization (EM) algorithm, that provides the joint ML estimate of X_K and θ. However, if the dimension of these two vectors is large compared to the dimension of Y_K, then the accuracy is deemed to degrade.

- If the sparsity structure is given, for instance parametrized with $\alpha_m^{-1} = 0$ for a set of indices m, then the EM method becomes more attractive. We will use this as a reference solution to get an upper bound on the possible performance.

- One can perform dedicated calibration experiments where the trajectory X_k is known, and estimate θ with a sparse method:

  - Assuming a Laplacian prior with precision α_m on each parameter leads to the popular l_1 optimization framework, that provides a regularized solution to the sparse estimation problem. Here α_m is a design parameter, which is a tricky user choice that affects the degree of regularization.

– Assuming a Gaussian prior with an unknown variance α_m is another approach. These variances can be estimated with the evidence approximation (EA) method.

What we propose is an iterative method consisting of two steps:

- The two vectors X_K and θ are estimated for a given value of α using the EM algorithm.
- The set of hyper-parameters α is estimated using the EM algorithm based on the EA principle.

2.2 Sensor and Motion Model

The measurement model for each sensor is assumed to be a known, nonlinear function $h(x_k, \theta)$ of state and bias. The measurement at time k, corresponding to state x_k is then defined as

$$y_k = h(x_k, \theta) + \nu_k, \tag{1}$$

where ν_k is a normally distributed zero mean additive noise with a known covariance matrix $R = B^{-1}$, being an inverse of precision matrix B.

In this paper, for simplicity, a case where bias vector enters the measurement function linearly will be considered. Then, the measurement function linear in θ is defined as

$$y_k = h^x(x_k) + h^\theta(x_k)\theta + \nu_k, \tag{2}$$

where $h^x(x_k)$ and $h^\theta(x_k)$ are terms that do not depend on θ. When the function is nonlinear, in order to transform it into a form linear in θ, one can use first order Taylor expansion around some known $\hat{\theta}$, and thus get

$$y_k = h(x_k, \hat{\theta}) + H_k^\theta(\theta - \hat{\theta}) + \nu_k, \tag{3}$$

where

$$H_k^\theta = \left.\frac{\partial h(x_k, \theta)}{\partial \theta}\right|_{\theta = \hat{\theta}} \tag{4}$$

is a Jacobian. In general it is assumed, that the target moves according to a general motion model, described by an equation

$$x_{k+1} = f(x_k, \eta_k), \tag{5}$$

where $f(x_k, \eta_k)$ is some transition function of a state x_k and a process noise η_k, with known distribution. For the purpose of this paper, a linear Constant Velocity (CV) model will be used, defined as

$$x_{k+1} = F_k x_k + G_k \eta_k, \tag{6}$$

where F_k and G_k are known matrices, and η_k is a normally distributed noise with zero mean and covariance matrix Q.

The task is to estimate the bias parameter vector θ in a situation where only the measurement set Y_K is available and the positions of the target X_K are unknown.

Since the problem of sensor network calibration might not always be well defined, due to limited observability or small number of measurements, the traditional maximum likelihood methods, as was mentioned, might not be sufficient and will provide unreliable results. The problem can be handled by applying some constraints on the estimates, which in the Bayesian framework is solved by assigning the prior. Choosing the right prior is also a problem itself, since it should utilize the sparsity of the parameter vector.

2.3 Expectation Maximization (EM) Algorithm

The ideas behind the EM, as applicable to our problem, are as follows. It is a two-step iterative method used for finding the maximum likelihood (ML) or the maximum a posteriori (MAP) estimates of parameters in a situation, when the model depends on latent (hidden and unknown) variables. In the considered case of parameter estimation, where based on set of measurements Y_K one wants to estimate θ, the latent variables are the sequence of states X_K. In the first step, called Expectation (E) step, one computes

$$Q(\theta, \hat{\theta}) = E_\theta\left[\ln p(X_K, Y_K|\theta)\middle| Y_K, \hat{\theta}\right] = \int \ln p(X_K, Y_K|\theta) p(X_K|Y_K, \hat{\theta}) dX_K, \tag{7}$$

that is a minimum variance estimate of a joint log-likelihood, using the available data set Y_K and an assumption $\hat{\theta}$ of the true parameter vector θ. The underlying idea is, that it should be much easier to maximize the complete data likelihood $p(X_K, Y_K|\theta)$, compared to direct maximization of $p(Y_K|\theta)$.

The second step is Maximization (M), where one calculates a new estimate of the parameter vector

$$\hat{\theta}^* = \arg\max_\theta Q(\theta, \hat{\theta}) \tag{8}$$

for the maximum likelihood case, and

$$\hat{\theta}^* = \arg\max_\theta \left(Q(\theta, \hat{\theta}) + \ln p(\theta)\right) \tag{9}$$

for the case of maximum a posteriori, where $p(\theta)$ is a prior over the bias parameter vector.

This estimate is then used as an input for the new iteration. As a result, the more iterations, the better approximation of the ML (or the MAP) estimates. It is also shown in the literature that the EM algorithm has a guaranteed convergence to at least a local minimum (Dempster et al. (1977), Bishop (2006)).

In order to obtain maximum a posteriori estimates, the prior needs to be assigned. Since it was assumed that the hyper-parameters of the prior are unknown, an evidence approximation method can be applied, according to the description provided in next section.

2.4 Evidence Approximation (EA)

General framework for the procedure of maximizing an evidence function – the likelihood of a data given set of hyper-patameters, called evidence procedure, can be found in MacKay (1992), Tipping (2001) or Gull and Skilling (1999).

In the general estimation framework the parameter vector is assumed to be a stochastic variable. The method consists of two subsequent levels of inference. In the first level one uses a Bayes' rule to find a posterior distribution for the bias parameters, given by

$$p(\theta|Y_K,\alpha) = \frac{p(Y_K|\theta,\alpha)p(\theta|\alpha)}{p(Y_K|\alpha)}, \tag{10}$$

where the prior over the parameter vector $p(\theta|\alpha)$ is governed by a set of hyper-parameters α, that are usually unknown and must also be inferred from the data. In a fully Bayesian framework though, in order to obtain estimates of θ, one shall integrate out the hyper-parameters in the following way

$$p(\theta|Y_K) = \int p(\theta|Y_K,\alpha)p(\alpha|Y_K)d\alpha, \tag{11}$$

where $p(\alpha|Y_K)$ is a posterior for the hyper-parameters obtained based on the set of measurements Y_K. Since it is usually problematic, one can make hare a simplification by assuming, that the posterior for hyper-parameters is sharply peaked around its estimate $\hat{\alpha}$, so $p(\alpha|Y_N) \approx \delta(\alpha - \hat{\alpha})$, and then use the approximation

$$p(\theta|Y_K) \simeq p(\theta|Y_K,\alpha)\Big|_{\alpha=\hat{\alpha}} = \frac{p(Y_K|\theta,\hat{\alpha})p(\theta|\hat{\alpha})}{p(Y_K|\hat{\alpha})}, \tag{12}$$

Thus the posterior for the parameter vector in (11) is assumed to be the same as in (10), but with fixed hyper-parameters $\alpha = \hat{\alpha}$.

In the second level of inference, set of unknown hyper-parameters is estimated. Using Bayes' rule again one can write

$$p(\alpha|Y_K) = \frac{p(Y_K|\alpha)p(\alpha)}{p(Y_K)}. \tag{13}$$

Assuming that the prior for the hyper-parameters is flat, $p(\alpha) = const$, in order to obtain new estimates of α one then only needs to maximize $p(Y_K|\alpha)$. The function is called an evidence. It is important to notice that it appears as a normalizing constant in (10), and is obtained by integrating out the parameter vector, as

$$p(Y_K|\alpha) = \int p(Y_K|\theta,\alpha)p(\theta|\alpha)d\theta. \tag{14}$$

Having the evidence function, one can finally obtain point estimates of hyper-parameters using direct maximization, as

$$\hat{\alpha} = \arg\max_{\alpha} p(Y_K|\alpha). \tag{15}$$

2.5 Two Latent Variables

This paper handles the case when both the state of the target X_N and bias vector θ are unknown. This situation therefore complicates the basic definition of the evidence procedure presented earlier, because one needs to make an extra integration of unknown state in both levels of inference. In the first step, the posterior for the parameters in (11) takes the form

$$p(\theta|Y_K) = \int p(X_K, \theta|Y_K) dX_K. \tag{16}$$

Analogously, a similar situation appears in the second step. The integration in (14), takes the following form

$$p(Y_K|\alpha) = \int p(X_K, Y_K|\alpha) dX_K = \iint p(X_K, Y_K, \theta|\alpha) dX_K d\theta. \tag{17}$$

One of the natural ways to handle this problem, due to presence of latent variables, is to use the EM algorithm. Next chapter presents the problem of application of the expectation maximization to the evidence procedure, where the evidence function is maximized in an iterative manner.

3 Sparse Calibration Algorithm

In order to infer the values of θ in a Bayesian framework, and to apply some constraints on the data, one needs to assign a prior. In this paper the following, zero mean Gaussian prior will be used, defined by

$$p(\theta|\alpha) = \mathcal{N}\left(\theta|0, A^{-1}\right), \tag{18}$$

where $A = \mathrm{diag}(\alpha_1 \dots \alpha_M)$ is a prior precision matrix and α_m is the m-th hyper-parameter.

As was mentioned before, there appears to be a problem how to assign the hyper-parameters to the prior. In the framework presented here they are assumed unknown and inferred from the data together with set of bias parameters, under the assumption that the prior over the hyper-parameters is constant (or non informative), so $p(\alpha) = const$.

Having defined all the necessary components and previously defined the problem in Section 2, we can proceed to the Bayesian inference of unknown variables in presence of latent variables X_K using expectation maximization procedure applied to the evidence procedure.

3.1 Application of the EM to the Problem

In order to provide a proper derivation of the algorithm it is necessary to start with the second level of inference. In this step the hyper-parameters of the prior for θ will be inferred from the data by maximizing the evidence function, according to the scheme presented in Section 2.4. The second level of inference requires

integrating out the parameter variable, as in (14). In presence of the latent variable X_N one get the following form of the evidence function (as in (17)) to be maximized

$$p(Y_K|\alpha) = \iint p(X_K, Y_K, \theta|\alpha) dX_K d\theta. \tag{19}$$

We can see here that maximization of the evidence can be solved iteratively by using the expectation maximization algorithm and treating both X_K and θ as latent variables. Thus having some initial estimate for the hyper-parameter $\hat{\alpha}$, we can proceed iteratively with both expectation and maximization steps.

Expectation step for the hyper-parameters: In this step, according to the scheme presented in Section 2.3, the expected value of a log joint-likelihood will be computed with respect to the posterior for both latent variables. The resulting $Q(\alpha, \hat{\alpha})$ function is then given by

$$\begin{aligned} Q(\alpha, \hat{\alpha}) &= E\Big[\ln p(X_K, Y_K, \theta|\alpha)\Big|Y_K, \hat{\alpha}\Big] \\ &= \iint \ln p(X_K, Y_K, \theta|\alpha) p(X_K, \theta|Y_K, \hat{\alpha}) dX_K d\theta. \end{aligned} \tag{20}$$

Decomposing the joint log likelihood one gets

$$\ln p(X_K, Y_K, \theta|\alpha) = \ln p(X_K, Y_K|\theta, \alpha) + \ln p(\theta|\alpha). \tag{21}$$

Under the assumption that the initial state distribution $p(x_1)$ is known, one can continue the decomposition as

$$\ln p(X_K, Y_K, \theta|\alpha) = \sum_{k=1}^{K} \ln p(y_k|x_k, \theta, \alpha) + \ln p(\theta|\alpha) + const. \tag{22}$$

Here one can immediately make an observation that the likelihood $p(y_k|x_k, \theta, \alpha)$ does not depend on α, so we can put all elements that do not depend on it as a constant ($const$), resulting in the following expression depending on the hyper-parameters

$$\ln p(X_K, Y_K, \theta|\alpha) = -\frac{1}{2} \ln |A| - \frac{1}{2} \theta^T A \theta + const. \tag{23}$$

Using the above log likelihood, the $Q(\alpha, \hat{\alpha})$ function in (20), after some simple rearranging, can be stated as

$$Q(\alpha, \hat{\alpha}) = -\frac{1}{2} \int \underbrace{(\ln |A| + \frac{1}{2} \theta^T A \theta)}_{①} \underbrace{\int p(X_K, \theta|Y_K, \hat{\alpha}) dX_K}_{②} d\theta + const. \tag{24}$$

Since ① does not depend on X_K, one only needs integrate out set of target states from the joint posterior $p(X_K, \theta|Y_K, \hat{\alpha})$. As one should immediately notice, it in fact directly corresponds to the first level of inference presented in latter part of Section 2.4, where one uses the assumption, that posterior for α is sharply peaked

around the hyper-parameter estimate. Then one can use the approximation as in (12), so (2) can be identified as

$$p(\theta|Y_K) \approx p(\theta|Y_K, \hat{\alpha}) = \int p(X_K, \theta|Y_K, \hat{\alpha}) dX_K, \tag{25}$$

and the $Q(\alpha, \hat{\alpha})$ function can be rewritten in a much simpler form of the following expectation

$$Q(\alpha, \hat{\alpha}) = -\frac{1}{2} \int (\ln|A| + \theta^T A\theta) p(\theta|Y_K, \hat{\alpha}) d\theta + const. \tag{26}$$

The problem to be solved at this stage is how to compute the posterior distribution of parameter vector $p(\theta|Y_K, \hat{\alpha})$, given set of measurements Y_K, in presence of the set of latent variables X_K while having the set of hyper-parameters $\alpha = \hat{\alpha}$ fixed. This fits again in a natural way into the expectation maximization procedure.

Using some initial value of θ, one can now continue with two consecutive steps of the algorithm – expectation step and maximization step, according to the scheme presented in Section 2.3.

Expectation step for the parameters: In this step one computes the expectation of a joint log-likelihood with respect to the posterior for latent variables, defined as

$$Q_{\text{MAP}}(\theta, \hat{\theta}) = Q(\theta, \hat{\theta}) + \ln p(\theta|\hat{\alpha}), \tag{27}$$

where

$$\begin{aligned} Q(\theta, \hat{\theta}) &= E\Big[\ln p(X_K, Y_K|\theta, \hat{\alpha}) \Big| Y_K, \hat{\theta}, \hat{\alpha}\Big] \\ &= \int \ln p(X_K, Y_K|\theta, \hat{\alpha}) p(X_K|Y_K, \hat{\theta}, \hat{\alpha}) dX_K \end{aligned} \tag{28}$$

and $\ln p(\theta|\hat{\alpha})$ is a log of the prior for θ. A posterior for X_K, given by $p(X_K|Y_K, \hat{\theta}, \hat{\alpha})$, is usually obtained from a smoothing algorithm (Rauch-Tung-Striebel for linear case or particle smoother for the nonlinear case etc.). Rewriting the joint log-likelihood into

$$\ln p(X_K, Y_K|\theta, \hat{\alpha}) = \ln p(Y_K|X_K, \theta, \hat{\alpha}) + \ln p(X_K|\theta, \hat{\alpha}), \tag{29}$$

and assuming again that initial target state $p(x_1)$ is known, the above can be rewritten again, keeping the terms that do not depend on θ as a constant, as

$$\ln p(X_K, Y_K|\theta, \hat{\alpha}) = \sum_{k=1}^{K} \ln p(y_k|x_k, \theta, \hat{\alpha}) + const. \tag{30}$$

Using (1) it can be further rewritten as

$$\ln p(X_K, Y_K|\theta, \hat{\alpha}) = -\frac{1}{2} \sum_{k=1}^{K} \Big(h^T(x_k, \theta) B h(x_k, \theta) - 2 y_k^T B h(x_k, \theta) \Big) + const. \tag{31}$$

Now, in order to take the expectation, as in (28), the measurement function will be linearized around $\hat{\theta}$ and the smoothed state estimate $\hat{x}_{k|K} = E\{x_k|Y_K, \hat{\theta}, \hat{\alpha}\}$, that is obtained from Extended Kalman Smoother using Rauch-Tung-Striebel equations. Taking the expectation of joint log-likelihood, keeping again only the terms depending on θ, and after some cancelations of terms linear in $x_k - \hat{x}_{k|K}$ (as their expected value vanishes as a direct result of linearization around smoothed state estimate), one can rewrite (28) as

$$Q(\theta, \hat{\theta}) = -\frac{1}{2}\sum_{k=1}^{K}\Big(-2(y_k - h(\hat{x}_{k|K}, \hat{\theta}) + H_k^{\theta}\hat{\theta})^T B H_k^{\theta}\theta + \theta^T (H_k^{\theta})B\theta\Big) + const. \quad (32)$$

Making use of the prior definition in (18), and keeping again only terms dependent on the parameter vector, the expectation of joint log-likelihood takes finally the form

$$Q_{\text{MAP}}(\theta, \hat{\theta}) = Q(\theta, \hat{\theta}) - \frac{1}{2}\theta^T \hat{A}\theta + const, \quad (33)$$

where $\hat{A} = \text{diag}(\hat{\alpha}_1 \dots \hat{\alpha}_M)$ is a diagonal matrix with estimates of hyper-parameters located on a diagonal.

Maximization step for the parameters: In this step one maximizes the $Q_{\text{MAP}}(\theta, \hat{\theta})$ function w.r.t. θ, as in (9). By using the property of symmetry of measurement precision matrix $B^T = B$, and solving the equation

$$\frac{\partial}{\partial\theta} Q_{\text{MAP}}(\theta, \hat{\theta}) = 0, \quad (34)$$

that yields the solution of the form

$$\hat{\theta}^* = \Sigma_{\theta}^{-1}\left(\sum_{k=1}^{K}(H_k^{\theta})^T B\Big(y_k - h(\hat{x}_{k|K}, \hat{\theta}) + H_k^{\theta}\hat{\theta},\Big)\right) \quad (35)$$

where Σ_θ is a corresponding estimate covariance matrix, defined as

$$\Sigma_{\theta} = \left(\sum_{k=1}^{K}(H_k^{\theta})^T B H_k^{\theta} + A\right)^{-1}. \quad (36)$$

Having obtained the estimate of θ, one can use it as a a starting point for the new iteration of the EM algorithm. At this step it is important to notice, that both the estimate and its covariance matrix form a Gaussian posterior for θ, with mean $\mu_\theta = \hat{\theta}^*$ and covariance matrix Σ_θ. From (25) we can then write

$$p(\theta|Y_K) \simeq p(\theta|Y_K, \hat{\alpha}) = \mathcal{N}(\theta|\mu_\theta, \Sigma_\theta). \quad (37)$$

Having estimated the posterior for parameter vector, which turned out to be equivalent to the first step of evidence procedure, one can now return to the second step, where the set of hyper-parameters is being finally inferred from the data.

MAXIMIZATION STEP FOR THE HYPER-PARAMETERS: Using the result from first level of inference for the posterior $p(\theta|Y_K, \hat{\alpha})$, having a Gaussian form given in (37) and a property of matrix trace $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$, the final form of the expectation from (26) can be rewritten as

$$Q(\alpha, \hat{\alpha}) = -\frac{1}{2}\ln|A| - \frac{1}{2}\mathrm{Tr}(A(\Sigma_\theta + \mu_\theta \mu_\theta^T)) + const. \tag{38}$$

In the maximization step the expected value of log joint-likelihood is maximized with respect to each i-th element of α. Solving the equation

$$\frac{\partial Q(\alpha, \hat{\alpha})}{\partial \alpha_m} = 0 \tag{39}$$

by also using the property of a trace, $\frac{\partial}{\partial A_{ij}}\mathrm{Tr}(AB) = B_{ji}$, gives the final result

$$\hat{\alpha}_i^* = \frac{1}{[\Sigma_\theta + \mu_\theta \mu_\theta^T]_{ii}} = \frac{1}{[\Sigma_\theta]_{ii} + [\mu_\theta]_i^2} = \frac{1}{[\Sigma_\theta]_{ii} + \hat{\theta}_i^2}, \tag{40}$$

where $[\cdot]_{ii}$ denotes i^{th} diagonal element, and $[\cdot]_i$ denotes i^{th} row/element.

Having computed new value of $\hat{\alpha}$, the iterative procedure is repeated until algorithm converges or assumed number of iterations was performed.

3.2 The Algorithm

The complete algorithm derived in a previous subSection is presented in Algorithm 1. After taking a deeper look, one can notice that it can be considered as a type of 'double-looped' EM algorithm. The basic principle is that given an initial value of hyper-parameter $\hat{\alpha}$ one computes expectation $Q(\alpha, \hat{\alpha})$ of the log of joint-likelihood – this step is here named E_1. The expectation for the parameters (here named E_2) is computed during steps from $2a$) to $2e$), using again the EM algorithm in the inner loop (steps a)–d)). The results obtained are used to compute $Q(\alpha, \hat{\alpha})$ in e). It is then maximized in the maximization step (respectively named M_2) in f). Having obtained a new estimate of the hyper-parameters, the outer loop is iterated again, until convergence.

The traditional approach to the parameter estimation using expectation maximization comes directly from the method definition, stating that EM is an iterative method to obtain a maximum likelihood estimates of the parameters in a presence of latent variables. The main task is to maximize the $Q(\theta, \hat{\theta})$ function. One can also extend the method, in case when the prior distribution over the parameters is known, so the EM algorithm will compute the maximum a posteriori, by maximizing the $Q_{\mathrm{MAP}}(\theta, \hat{\theta})$.

The algorithm presented in this paper extends the standard approach to the EM method, by also inferring the parameters of the prior. Having some initial values of the hyper-parameters $\hat{\alpha}$, in each iteration their new values are inferred according to (40). This simple modification has an advantage, that during the re-estimation of hyper-parameters the Occam's razor is employed, which in result

Algorithm 1 Sparse calibration algorithm using EM

1. Initialize the algorithm iterations at $n = 0$ with some arbitrary initial values $\hat{\theta}^{(n)} = \theta^0$ and $\hat{\alpha}^{(n)} = \alpha^0$.
2. For $n = 1 \dots N$:
   (a) Initialize algorithm with $\hat{\alpha} = \hat{\alpha}^{(n-1)}$

   FIRST LEVEL OF INFERENCE:
   (b) Initialize EM algorithm at $j = 0$ with initial values $\hat{\theta}^{(j)} = \hat{\theta}^{(n-1)}$
   (c) For $j = 1 \dots J$:
   EXPECTATION (E_1) STEP:
   - Assign $\hat{\theta} = \hat{\theta}^{(j-1)}$
   - Obtain set of state estimates $\{\hat{x}_{k|K}\}$ for $k = 1 : K$ using Extended Kalman Filter and Smoothing algorithm
   - Obtain $Q_{\text{MAP}}(\theta, \hat{\theta})$ function according to the (32)

   MAXIMIZATION (M_1) STEP:
   - Obtain new estimate of θ as in (35)
   - Assign new estimate, so thus $\hat{\theta}^{(j)} = \hat{\theta}^*$

   (d) If ($j == J$): assign $\hat{\theta}^{(n)} = \hat{\theta}^{(J)}$ and go to *d*),
   otherwise: increment $j = j + 1$ and go to *b*).

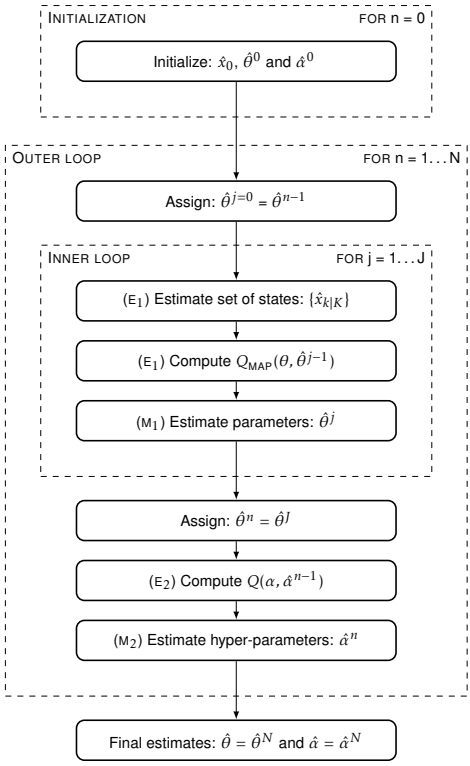
   SECOND LEVEL OF INFERENCE:
   (e) EXPECTATION (E_2) STEP:
   - Assign $\hat{\alpha} = \hat{\alpha}^{(n-1)}$
   - Compute $Q(\alpha, \hat{\alpha})$ as in (38)

   (f) MAXIMIZATION (M_2) STEP:
   - Obtain new estimate of α as in (40)
   - Assign new estimate, so thus $\hat{\alpha}^{(n)} = \hat{\alpha}^*$
3. If ($n == N$) or converged: end iterations,
   otherwise: increment $n = n + 1$ and continue.

exploits the sparsity feature of parameter vector. Next Section describes in detail how the Occam's razor is utilized and further in Section 4 all three methods are compared, both on simulated and real world scenario.

3.3 Interpretation of the Occam's Razor

As it was previously mentioned, the important feature of maximizing the evidence is that it implies an Occam's razor and allows us to find a compromise between model complexity (in our case the value of α representing directly a range of values that θ can take) and a best fit of θ to the data. Now the basics underlying the concept will be explained; an extended description can be found in MacKay (1992) or Bishop (2006).



(a) *Extended EM algorithm using evidence approximation.*

(b) *Traditional EM algorithm for ML/MAP estimates.*

Figure 1: *Comparison of the traditional approach to EM for ML (or MAP) estimates with the extended approach presented in the paper.*

Consider a simple, one dimensional example, where θ is a scalar. Assuming for simplicity that the prior over this parameter is uniformly distributed on some interval $\Delta\theta_{\text{PRIOR}}$, and by applying Laplace approximation to the evidence in (14) under an assumption, that the posterior is peaked around its most probable value $\hat{\theta}_{\text{MAP}}$, which holds in most cases, we get

$$p(Y_K|\alpha) \simeq p(Y_K|\hat{\theta}_{\text{MAP}}, \alpha) \times (\Delta\hat{\theta}_{\text{MAP}}/\Delta\theta_{\text{PRIOR}}). \tag{41}$$

By identifying certain terms of the equation, we can write

$$\textit{Evidence} \simeq \textit{Best fit likelihood} \times \textit{Occam's factor}. \tag{42}$$

It shows, that the expression for evidence is equal to the product of the best fit likelihood $p(Y_K|\hat{\theta}_{\text{MAP}}, \alpha)$ and the Occam's factor $\Delta\hat{\theta}_{\text{MAP}}/\Delta\theta_{\text{PRIOR}}$, which is represented by the ratio of the volumes of accessible values of θ for the posterior and prior. The Occam's factor is a penalty term measuring the complexity of the model.

The wider the $\Delta\theta_{\text{PRIOR}}$ is the variety of values the parameter can take is much larger, which makes model more complex. On the other hand, for the models finely tuned to the data, the value of $\Delta\hat{\theta}_{\text{MAP}}$ is much smaller, compared to the width of the prior, so it will imply larger penalty. Maximizing the evidence function lets us find values that provide an optimal trade-off between minimizing the misfit to the data and a model complexity measure, since the Occam's razor prefers its balanced value.

4 Simulation Results

To verify performance of the algorithm, two simulated scenarios were considered, where, N_s sensors, are randomly distributed over an area of 500×500 meters. Sensors are simple radar sensors, gathering range (in meter) and bearing measurements (in radians) corresponding to a single target. Bias parameters affect both position of the sensors and measurements. Additive position biases, corresponding to x- and y- axis are defined as θ_x and θ_y respectively; additive measurement biases, corresponding to range and bearing measurements, are defined as θ_r and θ_a respectively.

Simulations are performed in a Monte Carlo (MC) manner, where biases are randomly generated in each run, according to a zero-mean normal distribution with standard deviations σ_x, σ_y, σ_r and σ_a corresponding respectively to bias parameters. Bias vector is also assumed to be sparse, where s_f represents the sparsity factor - a fraction of non-zero elements and ranging in value from 0 to 1.

Table 1: *Parameters for both simulated scenarios*

	Scenario:	1	2
Parameter	Symbol	Value	Value
Number of measurements/sensor	K	100	25
Number of sensors	N_s	10 m	
Range measurement noise std	σ_y^r	10 m	
Angle measurement noise std	σ_y^a	10°	
Number of MC runs	N_{mc}	100	
Sparsity range	s_p	0 : 0.1 : 1	
Number of inner/outer loop iterations	I_{in}/I_{out}	1/50	
Initial position/velocity std	σ_p/σ_v	2.5m / 1m/s	
Bias: x/y position std	σ_x/σ_y	10m / 10m	
Bias: range/angle std	σ_r/σ_a	10m / 10°	

For the first scenario target moves with a sinusoidal trajectory, with a relatively high number of measurements, which should provide good observability of the parameters. In the second scenario, trajectory is much simpler, which should result in problems with observability of the bias parameters. Detailed description of both scenarios is presented in Table 1 together with an overview in Figure 2.

Measurement function (corresponding to the definition in (1)), used to gather the

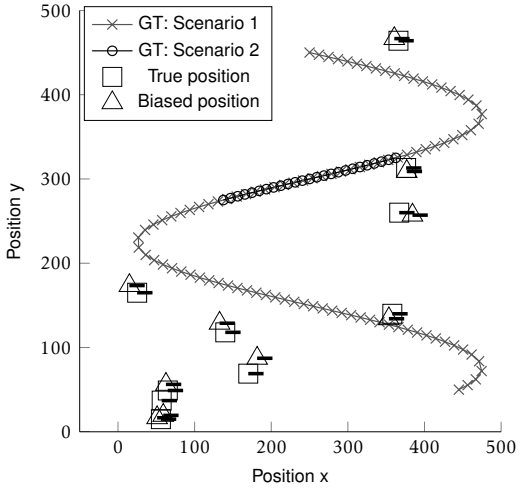


Figure 2: *General overview of the two simulated scenarios presenting ground truth (GT) and true and biased positions of the sensors.*

measurements, is defined for each radar sensor as

$$h(x_k, \theta) = \begin{bmatrix} h_1(x_k, \theta) \\ h_2(x_k, \theta) \end{bmatrix} = \begin{bmatrix} \sqrt{d_{x,k}^2 + d_{y,k}^2} + \theta_r \\ \arctan \frac{d_{y,k}}{d_{x,k}} + \theta_a \end{bmatrix}, \tag{43}$$

where

$$d_{x,k} = p_{x,k}^x - (p_x^s - \theta_x), \quad d_{y,k} = p_{y,k}^x - (p_y^s - \theta_y), \tag{44}$$

and where the target state vector $x_k = [p_{x,k}^x, p_{y,k}^x, v_{x,k}^x, v_{y,k}^x]^T$, with elements corresponding to x- and y-position and x- and y-velocity of the target at time step k respectively and p_x^s and p_y^s are positions of the sensor in x- and y-axis respectively.

Motion model (corresponding to (5)), used in order to obtain the set of state estimates $\{x_{k|K}\}$ for time steps $k = 1 \dots K$, is a standard, two dimensional Constant Velocity (CV) model, defined as

$$x_{k+1} = Fx_k + G\eta_k, \tag{45}$$

where η_k is a zero-mean normally distributed noise with a known 2×2 covariance matrix Q and

$$F = \begin{bmatrix} I_{2\times2} & TI_{2\times2} \\ 0_{2\times2} & I_{2\times2} \end{bmatrix}, \quad G = \begin{bmatrix} \frac{T^2}{2} I_{2\times2} \\ TI_{2\times2} \end{bmatrix}. \tag{46}$$

Each scenario was evaluated for all three algorithms mentioned before: for the standard EM algorithm computing maximum likelihood estimate (EM-ML; no prior considered), for the version of the EM algorithm computing the MAP esti-

mate (EM-MAP; true parameters of the distribution used to generate the biases were used as parameters of the prior) and for the newly proposed algorithm computing the MAP estimate (EM-EA), where hyper-prior parameters were computed iteratively during each iteration.

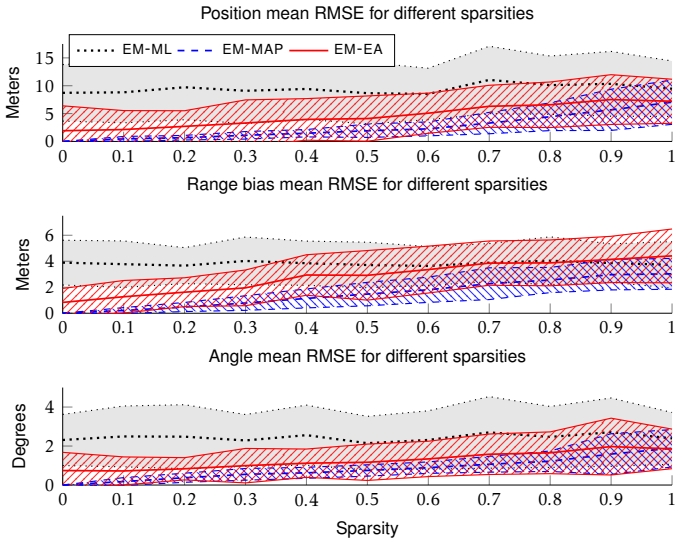
Results of MC runs are presented in Figure 3a for Scenario 1 and in Figure 3b for Scenario 2. As one can see, the proposed method performs much better compared to the EM method computing only the ML estimate. It comes naturally from the fact, that while computing the MAP estimate extra information is available in a form of the prior distribution. Additionally, the prior parameters are re-estimated in each iteration, as the second step of inference. Re-estimation allows to utilize the feature of the parameter vector sparsity through the Occam's razor, as was described in Section 3.3. One can thus observe, that the more sparse the parameter vector is, the better the method performs compared to the basic version of the EM algorithm. Application of the prior also helps to handle the problems of poor observability, as can be observed for Scenario 2. For that case computing the ML estimate does not provide reliable results at all, since the computations diverge just after few EM iterations. In opposite, the following situation does not occur for the proposed algorithm – it still provides reliable results.

As for the reference, both methods presented above are compared with the expectation maximization algorithm version using true prior for the parameter vector. Term *'true'* comes from the fact, that the prior used to obtain MAP estimate is the same as the one used to initially generate biases in the scenario. As should be expected, the method performance is the best among all three algorithms. It comes directly from the fact, that EM-MAP has a full initial information about the zero/non-zero parameters, which is incorporated in prior distribution. Analogous situation holds for the EM using evidence approximation, but in that case, that information is obtained during the second level of inference process. It is also visible, that the more sparse bias vector, the smaller RMS estimation error.

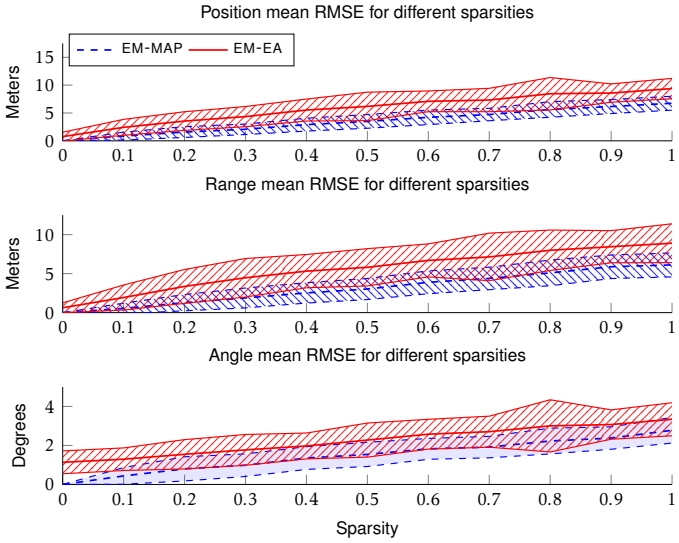
Figure 4 presents a result of one simulation for Scenario 1, comparing all three methods by presenting true and estimated values of a rather sparse ($s_f \simeq 0.4$) bias vector (upper plot) together with true and estimated standard deviations of the bias vector prior (bottom plot). As one can see, the proposed method evaluates the sparsity feature by detecting correctly parameters with zero/non-zero values. Method performance is thus very similar to the EM version computing the MAP estimate (with the prior containing full information about sparsity). On the other hand, simple EM algorithm assigned non-zero values to the almost all bias vector elements (since it does not evaluate the sparsity feature), finally causing much larger estimation error.

5 Real Data Experiment

To further verify performance of the method, all three algorithms were tested on real world scenario, presented in Figure 5, where $N_s = 9$ acoustic sensors



(a) *Results for Scenario 1.*



(b) *Results for Scenario 2.*

Figure 3: *Monte Carlo results for both simulated scenarios (mean and one standard deviation).*

(microphones) were observing one moving target.

Each sensor collects an acoustic signal, where the amplitude is measured (in Volts). RMS of the measured signal, over selected period, y_{RMS}, is then converted

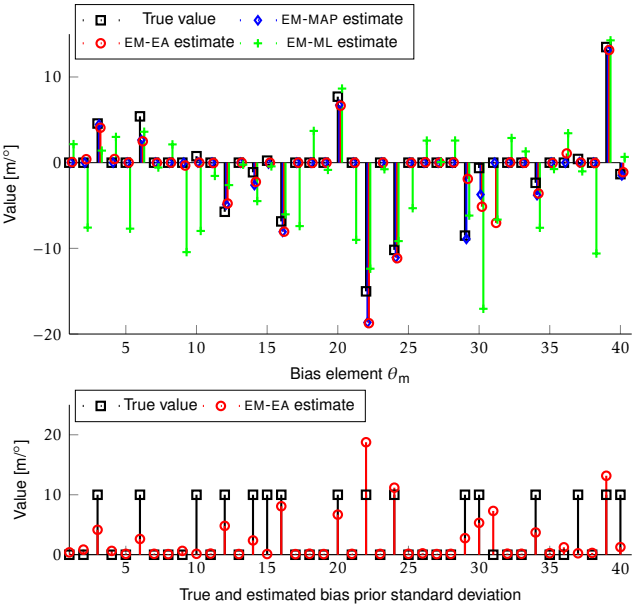


Figure 4: *Estimation results for one Monte Carlo simulation of Scenario 1.*

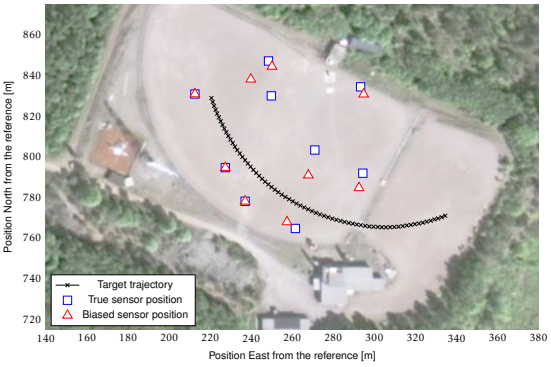


Figure 5: *Real scenario overview with target trajectory and sensor positions.*

into *sound pressure* measurements p_{RMS}, measured in Pascals, using a conversion factor specific to each sensor. The sound pressure p decays inversely proportionally to the distance to the source r, according to

$$p \propto r^{-1}. \tag{47}$$

In order to connect the position of the target with measures signal, one needs to introduce two quantities: *sound pressure level* (SPL) and *sound power level* (SWL). The first one (also known as *sound level*) is a logarithmic measure of an effective sound pressure, measured by the sensor, relative to a reference value $p_{\text{REF}} = 20\mu Pa$.

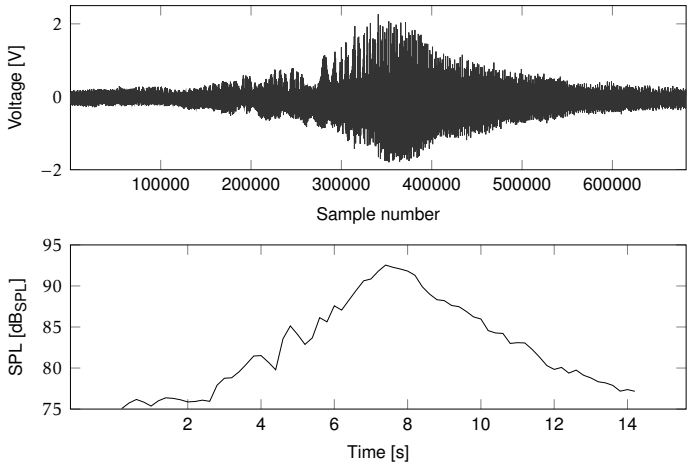


Figure 6: *Sound wave and computed SPL for one of the sensors.*

It is defined as

$$L^P = 20 \log_{10} \left(\frac{p_{\text{RMS}}}{p_{\text{REF}}} \right). \tag{48}$$

Sound pressure level is measured in a unit called dB_{SPL}, that is not to be confused with dB, since it is not recognizable SI unit. Using the reference pressure, a sound pressure of $1Pa$ corresponds to $94dB_{\text{SPL}}$ of sound level. Second measure, the sound power level (also known as an *acoustic power level*), is also a logarithmic measure. It is measured relative to a reference sound power ($P_0 = 10^{-12}W$, the reference sound power in the air) and computed as

$$L^W = 10 \log_{10} \left(\frac{P}{P_0} \right). \tag{49}$$

SWL unit is dB_{SWL}, and is also not recognizable as a SI unit.

Opposite to the sound pressure, sound power is not distance dependent, since it strictly characterizes the sound source. It represents the total power produced by the source in all dimensions, while the sound pressure is a measurement at a point near the source. The relation between log measures (SPL and SWL), for a free field source, is given by

$$L^P = L^W + 10 \log_{10} \left(\frac{S_0}{4\pi r^2} \right), \tag{50}$$

where $S_0 = 1m^2$. Recorded sound wave and computed sound pressure level are presented in Figure 6.

Having defined this relation, one can introduce a sound pressure level measurement function (corresponding to (1) and (48)), defined for each acoustic sensor

as

$$h(x_k, \theta) = (1 - \theta_G)L_k^w + 10\log_{10}(4\pi) - 10\log_{10}(r_k^2), \tag{51}$$

where we make use of an augmented state vector

$$x_k = [p_{x,k}^x, p_{y,k}^x, v_{x,k}^x, v_{y,k}^x, L_k^w]^T, \tag{52}$$

with L_k^w being the SWL emitted by a target at time k. Having previously defined $d_{x,k}$ and $d_{y,k}$ in (44), the

$$r_k^2 = d_{x,k}^2 + d_{y,k}^2 \tag{53}$$

is a squared distance to the target at time k. For acoustic sensors 3 different biases are distinguished, $\theta_x, \theta_y, \theta_G$, corresponding to x-/y- additive position bias and microphone gain miscalibration.

To describe the target motion, again CV model is used, as in (45), but in this case, in order to handle extra emitted power variable, model matrices had to be augmented, as

$$F = \text{blkdiag}(F, 1), \quad G = \text{blkdiag}(G, 1). \tag{54}$$

Since the scenario was pre-calibrated using very accurate differential GPS, in order to check performance of the algorithm one had to manually add random biases to certain sensors. Analogously to simulated scenarios, parameters are presented in Table 2. For sensor 1 the θ_y was fixed as 15m. In order to provide observability for the EM with ML, parameters for three sensor (3, 6 and 7 corresponding to bias elements $\{\theta_m\}_{m=\{7-9,16-21\}}$) had to be fixed a priori to zero.

Table 2: *Parameters for real world scenario*

Parameter	Symbol	Value
Number of measurements/sensor	K	81
Sampling rate	F_S	48 kHz
Number of sensors/fixed sensors	N_s/N_f	9/3
Measurement noise std	σ_y	2 dB_{SPL}
Number of MC runs	N_{mc}	50
Sparsity range	s_p	1
Number of inner/outer loop iterations	I_{in}/I_{out}	4/25
Initial position/velocity std	σ_p/σ_v	1m / 1m/s
Initial emitted power mean/std	L^w/σ_w	102.5/1 dB_{SPL}
Bias: x/y/gain std	$\sigma_x/\sigma_y/\sigma_G$	$7.5m$/7.5m/0.05
Bias: gain std	σ_g	0.05

Results for 50 Monte Carlo runs are presented in Table 3. Convergence result for fixed bias of Sensor 1 is presented in Figure 8, followed by one sample MC run estimation result presented in Figure 7.

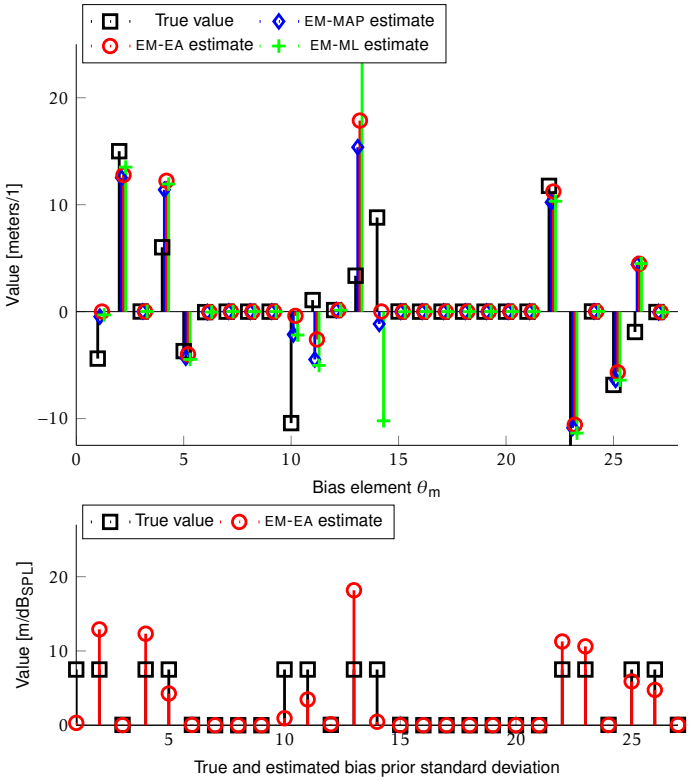


Figure 7: *Single MC run for the real world scenario, where three bias vector elements* $\{\theta_m\}_{m=\{7-9,16-21\}}$*, were manually fixed.*

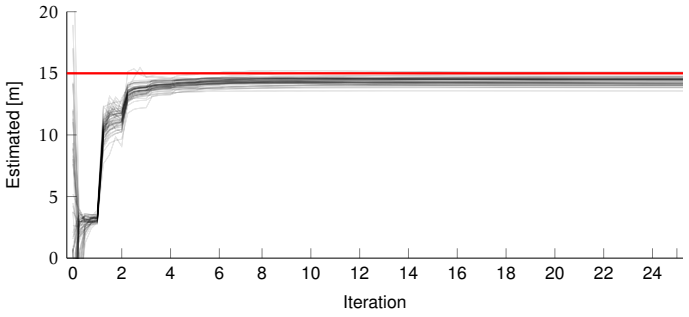


Figure 8: *Convergence of the* y *position bias* θ_y *for Sensor 1.*

6 Conclusions

In this paper a novel method for simultaneous tracking and calibration in sensor networks, utilizing the feature of sparsity of estimated parameter vector, was pre-

Table 3: *RMS error for the real world scenario*

Bias	Prior	EM-ML	EM-MAP	EM-EA
Position [m]	7.50	7.510 ± 4.486	5.276 ± 2.934	6.004 ± 3.125
Gain	0.05	0.009 ± 0.002	0.008 ± 0.001	0.008 ± 0.002

sented. New approach to the expectation maximization algorithm using evidence approximation, providing improved results, was compared with traditional approach to EM computing ML/MAP estimate. It was shown that method provides reliable results using only one target and limited number of measurements, and also in most cases correctly detects zero/non-zero bias parameters.

Acknowledgment

The research leading to these results was funded by EUs 7^{th} Framework Program under grant agreement no 238710 and was carried out in the MC IMPULSE project: `http://mcimpulse.isy.liu.se/`. Authors would also like to thank Lic. Zoran Sjanic for plenty of fruitful discussions regarding the derivation of the algorithm.

Bibliography

James O Berger. *Statistical decision theory and Bayesian analysis*. Springer, 1985.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

S Dhar. Application of a recursive method for registration error correction in tracking with multiple sensors. In *American Control Conference, 1993*, pages 869–874. IEEE, 1993.

SF Gull and J Skilling. Quantified Maximum Entropy MemSys5 Users' Manual. *Maximum Entropy Data Consultants Ltd., Suffolk*, 1999.

M Ignagni. An alternate derivation and extension of Friendland's two-stage Kalman estimator. *Automatic Control, IEEE Transactions on*, 26(3):746–750, 1981.

Sun Yuan Kung, Man-Wai Mak, and Shang-Hung Lin. *Biometric authentication: a machine learning approach*. Prentice Hall Professional Technical Reference, 2005.

Zhenhua Li, H Leung, and Liang Tao. Simultaneous registration and fusion of radar and ESM by EM-EKS. In *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, pages 1130–1134. IEEE, 2010.

David JC MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

Nickens Okello and Branko Ristic. Maximum likelihood registration for multiple dissimilar sensors. *Aerospace and Electronic Systems, IEEE Transactions on*, 39 (3):1074–1083, 2003.

Thomas B Schön, Adrian Wills, and Brett Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011.

Egils Sviestins. Online bias estimation for multisensor tracking. In *Information, Decision and Control, 1999. IDC 99. Proceedings. 1999*, pages 221–226. IEEE, 1999.

M. Syldatk and F. Gustafsson. Expectation maximization algorithm for simultaneous tracking and sparse calibration of sensor networks. In *IEEE Transactions on Signal Processing (TSP)*. To be submitted.

Michael E Tipping. Sparse Bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

Jaco Vermaak, Simon Maskell, and Mark Briers. Online sensor registration. In *Aerospace Conference, 2005 IEEE*, pages 2117–2125. IEEE, 2005.

David P Wipf and Bhaskar D Rao. Sparse Bayesian learning for basis selection. *Signal Processing, IEEE Transactions on*, 52(8):2153–2164, 2004.

Licentiate Theses
Division of Automatic Control
Linköping University

P. Andersson: Adaptive Forgetting through Multiple Models and Adaptive Control of Car Dynamics. Thesis No. 15, 1983.

B. Wahlberg: On Model Simplification in System Identification. Thesis No. 47, 1985.

A. Isaksson: Identification of Time Varying Systems and Applications of System Identification to Signal Processing. Thesis No. 75, 1986.

G. Malmberg: A Study of Adaptive Control Missiles. Thesis No. 76, 1986.

S. Gunnarsson: On the Mean Square Error of Transfer Function Estimates with Applications to Control. Thesis No. 90, 1986.

M. Viberg: On the Adaptive Array Problem. Thesis No. 117, 1987.

K. Ståhl: On the Frequency Domain Analysis of Nonlinear Systems. Thesis No. 137, 1988.

A. Skeppstedt: Construction of Composite Models from Large Data-Sets. Thesis No. 149, 1988.

P. A. J. Nagy: MaMiS: A Programming Environment for Numeric/Symbolic Data Processing. Thesis No. 153, 1988.

K. Forsman: Applications of Constructive Algebra to Control Problems. Thesis No. 231, 1990.

I. Klein: Planning for a Class of Sequential Control Problems. Thesis No. 234, 1990.

F. Gustafsson: Optimal Segmentation of Linear Regression Parameters. Thesis No. 246, 1990.

H. Hjalmarsson: On Estimation of Model Quality in System Identification. Thesis No. 251, 1990.

S. Andersson: Sensor Array Processing; Application to Mobile Communication Systems and Dimension Reduction. Thesis No. 255, 1990.

K. Wang Chen: Observability and Invertibility of Nonlinear Systems: A Differential Algebraic Approach. Thesis No. 282, 1991.

J. Sjöberg: Regularization Issues in Neural Network Models of Dynamical Systems. Thesis No. 366, 1993.

P. Pucar: Segmentation of Laser Range Radar Images Using Hidden Markov Field Models. Thesis No. 403, 1993.

H. Fortell: Volterra and Algebraic Approaches to the Zero Dynamics. Thesis No. 438, 1994.

T. McKelvey: On State-Space Models in System Identification. Thesis No. 447, 1994.

T. Andersson: Concepts and Algorithms for Non-Linear System Identifiability. Thesis No. 448, 1994.

P. Lindskog: Algorithms and Tools for System Identification Using Prior Knowledge. Thesis No. 456, 1994.

J. Plantin: Algebraic Methods for Verification and Control of Discrete Event Dynamic Systems. Thesis No. 501, 1995.

J. Gunnarsson: On Modeling of Discrete Event Dynamic Systems, Using Symbolic Algebraic Methods. Thesis No. 502, 1995.

A. Ericsson: Fast Power Control to Counteract Rayleigh Fading in Cellular Radio Systems. Thesis No. 527, 1995.

M. Jirstrand: Algebraic Methods for Modeling and Design in Control. Thesis No. 540, 1996.

K. Edström: Simulation of Mode Switching Systems Using Switched Bond Graphs. Thesis No. 586, 1996.

J. Palmqvist: On Integrity Monitoring of Integrated Navigation Systems. Thesis No. 600, 1997.

A. Stenman: Just-in-Time Models with Applications to Dynamical Systems. Thesis No. 601, 1997.

M. Andersson: Experimental Design and Updating of Finite Element Models. Thesis No. 611, 1997.

U. Forssell: Properties and Usage of Closed-Loop Identification Methods. Thesis No. 641, 1997.

M. Larsson: On Modeling and Diagnosis of Discrete Event Dynamic systems. Thesis No. 648, 1997.

N. Bergman: Bayesian Inference in Terrain Navigation. Thesis No. 649, 1997.

V. Einarsson: On Verification of Switched Systems Using Abstractions. Thesis No. 705, 1998.

J. Blom, F. Gunnarsson: Power Control in Cellular Radio Systems. Thesis No. 706, 1998.

P. Spångéus: Hybrid Control using LP and LMI methods – Some Applications. Thesis No. 724, 1998.

M. Norrlöf: On Analysis and Implementation of Iterative Learning Control. Thesis No. 727, 1998.

A. Hagenblad: Aspects of the Identification of Wiener Models. Thesis No. 793, 1999.

F. Tjärnström: Quality Estimation of Approximate Models. Thesis No. 810, 2000.

C. Carlsson: Vehicle Size and Orientation Estimation Using Geometric Fitting. Thesis No. 840, 2000.

J. Löfberg: Linear Model Predictive Control: Stability and Robustness. Thesis No. 866, 2001.

O. Härkegård: Flight Control Design Using Backstepping. Thesis No. 875, 2001.

J. Elbornsson: Equalization of Distortion in A/D Converters. Thesis No. 883, 2001.

J. Roll: Robust Verification and Identification of Piecewise Affine Systems. Thesis No. 899, 2001.

I. Lind: Regressor Selection in System Identification using ANOVA. Thesis No. 921, 2001.

R. Karlsson: Simulation Based Methods for Target Tracking. Thesis No. 930, 2002.

P.-J. Nordlund: Sequential Monte Carlo Filters and Integrated Navigation. Thesis No. 945, 2002.

M. Östring: Identification, Diagnosis, and Control of a Flexible Robot Arm. Thesis No. 948, 2002.

C. Olsson: Active Engine Vibration Isolation using Feedback Control. Thesis No. 968, 2002.

J. Jansson: Tracking and Decision Making for Automotive Collision Avoidance. Thesis No. 965, 2002.

N. Persson: Event Based Sampling with Application to Spectral Estimation. Thesis No. 981, 2002.

D. Lindgren: Subspace Selection Techniques for Classification Problems. Thesis No. 995, 2002.

E. Geijer Lundin: Uplink Load in CDMA Cellular Systems. Thesis No. 1045, 2003.

M. Enqvist: Some Results on Linear Models of Nonlinear Systems. Thesis No. 1046, 2003.

T. Schön: On Computational Methods for Nonlinear Estimation. Thesis No. 1047, 2003.

F. Gunnarsson: On Modeling and Control of Network Queue Dynamics. Thesis No. 1048, 2003.

S. Björklund: A Survey and Comparison of Time-Delay Estimation Methods in Linear Systems. Thesis No. 1061, 2003.

M. Gerdin: Parameter Estimation in Linear Descriptor Systems. Thesis No. 1085, 2004.

A. Eidehall: An Automotive Lane Guidance System. Thesis No. 1122, 2004.

E. Wernholt: On Multivariable and Nonlinear Identification of Industrial Robots. Thesis No. 1131, 2004.

J. Gillberg: Methods for Frequency Domain Estimation of Continuous-Time Models. Thesis No. 1133, 2004.

G. Hendeby: Fundamental Estimation and Detection Limits in Linear Non-Gaussian Systems. Thesis No. 1199, 2005.

D. Axehill: Applications of Integer Quadratic Programming in Control and Communication. Thesis No. 1218, 2005.

J. Sjöberg: Some Results On Optimal Control for Nonlinear Descriptor Systems. Thesis No. 1227, 2006.

D. Törnqvist: Statistical Fault Detection with Applications to IMU Disturbances. Thesis No. 1258, 2006.

H. Tidefelt: Structural algorithms and perturbations in differential-algebraic equations. Thesis No. 1318, 2007.

S. Moberg: On Modeling and Control of Flexible Manipulators. Thesis No. 1336, 2007.

J. Wallén: On Kinematic Modelling and Iterative Learning Control of Industrial Robots. Thesis No. 1343, 2008.

J. Harju Johansson: A Structure Utilizing Inexact Primal-Dual Interior-Point Method for Analysis of Linear Differential Inclusions. Thesis No. 1367, 2008.

J. D. Hol: Pose Estimation and Calibration Algorithms for Vision and Inertial Sensors. Thesis No. 1370, 2008.

H. Ohlsson: Regression on Manifolds with Implications for System Identification. Thesis No. 1382, 2008.

D. Ankelhed: On low order controller synthesis using rational constraints. Thesis No. 1398, 2009.

P. Skoglar: Planning Methods for Aerial Exploration and Ground Target Tracking. Thesis No. 1420, 2009.

C. Lundquist: Automotive Sensor Fusion for Situation Awareness. Thesis No. 1422, 2009.

C. Lyzell: Initialization Methods for System Identification. Thesis No. 1426, 2009.

R. Falkeborn: Structure exploitation in semidefinite programming for control. Thesis No. 1430, 2010.

D. Petersson: Nonlinear Optimization Approaches to $\mathcal{H}_2$-Norm Based LPV Modelling and Control. Thesis No. 1453, 2010.

Z. Sjanic: Navigation and SAR Auto-focusing in a Sensor Fusion Framework. Thesis No. 1464, 2011.

K. Granström: Loop detection and extended target tracking using laser data. Thesis No. 1465, 2011.

J. Callmer: Topics in Localization and Mapping. Thesis No. 1489, 2011.

F. Lindsten: Rao-Blackwellised particle methods for inference and identification. Thesis No. 1480, 2011.

M. Skoglund: Visual Inertial Navigation andCalibration. Thesis No. 1500, 2011.

S. Khoshfetrat Pakazad: Topics in Robustness Analysis. Thesis No. 1512, 2011.

P. Axelsson: On Sensor Fusion Applied to Industrial Manipulators. Thesis No. 1511, 2011.

A. Carvalho Bittencourt: On Modeling and Diagnosis of Friction and Wear in Industrial Robots. Thesis No. 1516, 2012.

P. Rosander: Averaging level control in the presence of frequent inlet flow upsets . Thesis No. 1527, 2012.

N. Wahlström: Localization using Magnetometers and Light Sensors. Thesis No. 1581, 2013.
R. Larsson: System Identification of Flight Mechanical Characteristics. Thesis No. 1599, 2013.
Y. Jung: Estimation of Inverse Models Applied to Power Amplifier Predistortion. Thesis No. 1605, 2013.