Thesis for the Degree of Doctor of Philosophy

Bayesian Inference for Automotive Applications

Maryam Fatemi

CHALMERS

Department of Signals and Systems
Chalmers University of Technology
Göteborg, Sweden 2016

Bayesian Inference for Automotive Applications
Maryam Fatemi
ISBN 978-91-7597-438-5

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 4119
ISSN 0346-718X

Department of Signals and Systems
Signal Processing Group
Chalmers University of Technology
SE–412 96 Göteborg, Sweden

Typeset by the author using LaTeX.

Chalmers Reproservice
Göteborg, Sweden 2016

What I want does exist if I dare to find it.

Oranges are not the only fruit
Jeanette Winterson

To Roshanak, Ali , Mani
and my beloved
Aidin

Abstract

Environment perception is an important aspect of modern automated systems. The perception consists of fusing information from different sensors to estimate variables which provide a description of the scene. The main contributions of this thesis are in multiple object tracking and mapping, and nonlinear filtering methods for road geometry estimation, which are important topics in the design of advanced driver assistance systems and self-driving cars.

Road geometry estimation is required in many advanced driver assistance systems. In this thesis, we utilize non-linear filtering methods to perform long-range road geometry estimation by fusing measurements from different sensors. In this context, the process model describes the time evolution of the road state and the measurement model establishes the relationship between the sensor measurements and the road state. We parameterize our road model such that the time evolution of the state follows the manner by which roads are built and design a filtering algorithm to estimate it.

Sensor maps provide a description of the environment as seen through the lens of a sensor. These maps describe the measurement distribution of an environment as a function of the sensor position and are used to perform localization. In this thesis, we model the prior of a radar map by a Poisson process which allows us to incorporate the uncertainties in the number of landmarks, their states and data association hypotheses, into the model. We derive the exact theoretical batch multi-object posterior density of the map and use Gibbs sampling method to approximate the posterior.

Extended object tracking is an important problem in the context of multiple-object tracking which arises, for example, in tracking using automotive radars. We present a Poisson multi-Bernoulli mixture conjugate prior for multiple extended object tracking. Since the resulting posterior is intractable, approximations are required to obtain a feasible algorithm. We present two tractable solutions, one based on the full PMBM posterior and one based on approximating the PMBM by a PMB process. Multiple practical challenges arise in connection to the PMB filter, to which we provide pragmatic solutions that yield an efficient and tractable algorithm.

Acknowledgements

The PhD time at Chalmers has been a journey rich with valuable experiences and fun memories, all of which has a special place in my heart. I would like to express my gratitude to Prof. Mats Viberg and Prof. Tomas McKelvey for giving me the opportunity to join the signal processing group and for creating a warm and friendly environment for the group.

My deepest thanks goes to my supervisor Lennart Svensson for his support, valuable academic and non-academic discussions, and constructive feedbacks. I appreciate your passion for teaching and enthusiasm towards life. Many thanks to my co-supervisers Karl Granström and Lars Hammarstrand for their support, interesting discussions and timely feedbacks. Working with you has been a valuable experience for me.

Thanks to all my friends, current and former colleagues at Signals and Systems department, signal processing group and Volvo. Special thanks to my friend and former colleague, Ángel, for proof reading this thesis.

Thanks to my Iranian friends, Sahar, Ali, Mitra, Negar, Roozbeh, Azita, Bahram, Sadegh, Abbas, Pegah and Samar. Thanks for reminding me of the sounds and tastes of Iran. Your friendship is valuable to me. Living away from Iran is much easier with you guys around.

To my family in Europe, Vaji, Shayesteh, Jaleh and Javad thanks for your hospitality. There are places in Uppsala, Hamburg and Wuppertal that remind me of my parents'.

To Shohreh, Peymon and Azin, thanks for your warm hearts and never ending support.

My deepest gratitude goes to Roshanak, Ali and Mani for all their support and unconditional love. You are a constant source of energy and inspiration to me.

To my dearest Aidin, you are the best company I could have asked for. Words can not express what you mean to me. Thanks for all your love and support through thin and thick.

Maryam Fatemi
Göteborg, August 2016

List of Publications

This thesis is based on the work contained in the following papers which are also included in the list of references as papers [1–7]:

Paper I [1]

M. Fatemi, L. Svensson, M. Morelande and L. Hammarstrand, "A study of MAP estimation techniques for nonlinear filtering", in 15th International Conference on Information Fusion, Singapore, July 2012, pp. 1058–1065.

Paper II [2]

M. Fatemi, L. Hammarstrand, L. Svensson and ÁF García-Fernández, "Road geometry estimation using a precise clothoid road model and observations of moving vehicles", in 17th International Conference on Intelligent Transportation Systems, Qingdao, China, Oct. 2014, pp. 238–244 .

Paper III [3]

L. Hammarstrand, M. Fatemi, ÁF García-Fernández and L. Svensson, "Long-Range Road Geometry Estimation Using Moving Vehicles and Roadside Observations", Accepted for publication in IEEE Trans. Intell. Transp. Syst.

Paper IV [4]

M. Fatemi, L. Svensson, L. Hammarstrand and M. Lundgren, "Variational Bayesian EM for SLAM", in 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, Cancun, Mexico, Dec. 2015, pp. 501–504.

Paper V [5]

K. Granström, M. Fatemi and L. Svensson, " Poisson multi-Bernoulli

conjugate prior for estimation of both detected and undetected extended objects", Submitted to IEEE Trans. Signal Process.

Paper VI [6]
M. Fatemi, K. Granström, L. Svensson, F. J. R. Ruiz and L. Hammarstrand, "Poisson Multi-Bernoulli Radar Mapping Using Gibbs Sampling", Submitted to IEEE Trans. Signal Process.

Paper VII [7]
M. Fatemi, L. Svensson and K. Granström, "Poisson multi-Bernoulli filter for extended object tracking", to be submitted.

Other publications [8–10]

K. Granström, M. Fatemi and L. Svensson, "Gamma Gaussian inverse-Wishart Poisson multi-Bernoulli Filter for Extended Target Tracking", in 19th International Conference on Information Fusion, Germany, July 2016.

ÁF García-Fernández, L. Hammarstrand, M. Fatemi and L. Svensson, "Bayesian Road Estimation Using Onboard Sensors", in IEEE Trans. Intell. Transp. Syst. 2014.

Maryam Fatemi, Hamidreza Amindavar, James A. Ritcey, "Noise Reduction via Harmonic Estimation in Gaussian and non-Gaussian Environments", in Signal Process., 2010.

Contents

Part I

Introductory Chapters

Chapter 1

Introduction

This chapter depicts the research topics of the thesis against the bigger scene of the related fields. Motivation and the main focus of the thesis are discussed in Section 1.1, and main contributions are described in Section 1.2. We discuss research ideas for future extensions in Section 1.3, and the outline of the thesis is presented in Section 1.4.

1.1 Motivation and main focus

According to the World Health Organization (WHO), every year approximately 1.25 million people die as a result of a road traffic crash. A further 20 to 50 million people suffer non-fatal injuries, among those many will suffer a form of disability as a result of their injury. Road traffic injuries cause considerable economic burden (cost of treatment, reduced/lost productivity, etc) to victims, their families, and to nations as a whole. The scale of the problem is such that the WHO has announced 2011-2020 the decade of action on road safety [11].

In recent years, the automotive and technology industry together with their academic partners have joined forces to design safer and smarter vehicles. New modern vehicles are equipped with various safety and comfort functionalities that are designed to support the driver not only in critical situations but also in mundane driving tasks, such as driving in a heavy traffic.

One of the fastest growing areas of research is the development of self-driving vehicles. In the second half of 2015, Tesla Motors began to allow owners to switch on its Autopilot mode. Already in 2017, Volvo is going to put 100 self-driving cars in the hands of costumers to drive them in Gothenburg, Sweden. The expectations regarding the benefit of this technology are very high. For example, Google claims that its self-driving car technology

could save approximately 30,000 lives and prevent nearly 2 million additional injuries each year in the U.S. alone. Google also claims that it can reduce accident-related expenses by at least $400 billion a year in the U.S., see for example [12].

Apart from saving lives and money, some analysts believe that this technology has the capacity to cause fundamental changes to our behaviour towards car ownership. Currently, cars are typically parked 95 percent of the time. This is clearly a large waste of resources (dedicated spot, household income, etc.). As opposed to the current situation, ordering a self-driving car when needed and freeing that resource when no longer required, increases the usage rate, saves household income and frees much needed space in and around the cities for necessary developments. For more detailed discussions regarding the pros and cons of self-driving cars see [13].

Both driver assistance systems and self-driving vehicles rely on accurate perception of their environment as well as an accurate map of the environment. In both applications the quest for improved accuracy is far from over. Environment perception can be performed by fusing information received from sensors mounted on the host vehicle as well as information from other vehicles, thanks to modern communication systems. The process of fusing information from noisy observations from different sensors is a major theme in this thesis, and a problem which is commonly referred to as sensor fusion. The objective in sensor fusion is often to estimate a state variable that may describe, e.g.:

- the geometry of the road as seen by the vehicle,
- the number of and the distance to other vehicles on the road,
- the number, position and extent of stationary objects in the environment, etc.

Bayesian inference is one of the techniques that can be used to perform sensor fusion, and it is particularly attractive because it provides a systematic approach to incorporate model uncertainties into estimation. More specifically, Bayesian inference methods provide a framework for propagating uncertainties through time and estimating the quantities of interest as well as the measure of reliability for each estimate, based on the inherent uncertainty of the system.

This thesis focuses on Bayesian inference methods that can provide solutions for advanced driver assistance systems and self-driving vehicles. More specifically, the thesis covers two aspects. The first is the use of nonlinear filtering techniques for road geometry estimation, and the second is mapping and tracking of multiple extended objects, taking into account the

uncertainties in the number of present objects and data association between measurements and objects. Most of this work was carried out within the Non-Hit Car and Truck (NHCT) project and was supported by VINNOVA.

1.2 Contributions of The Appended Papers

Paper I [1]: A study of MAP estimation techniques for nonlinear filtering
A new maximum a posteriori (MAP) based nonlinear filter is developed and compared to existing MAP-based filters and nonlinear filters. A situation for which MAP estimation techniques perform better than linear minimum mean squared error (LMMSE) based filters is illustrated. MAP-based filters are often iterative algorithms which are more computationally complex than LMMSE-based filters. The computational complexity results from the iterative optimization algorithm. A dimension reduction algorithm is proposed which can reduce the added computational burden of the optimization algorithm.

I derived the new methods and carried out the simulations under the supervision of the co-authors.

Paper II [2]: Clothoid-Based Road Geometry Estimation Using Moving Objects
We present a road geometry estimation algorithm based on Bayesian inference. A new road model which describes the road as clothoid segments, connected such that the whole curve maintains G^2-continuity, is developed. The road geometry is considered as the state of a dynamic system estimated from the measurements of the lane markings received from a camera and measurements of the leading vehicles received from a radar-camera fusion system. We consider two modes for the movement of the leading vehicles: 1) following lane and 2) changing lane, and use a multiple-model filter, based on these two modes, to update the posterior density of the road state. We evaluate the road geometry algorithm on real data collected from highways.

This work was carried out in close cooperation with the co-authors. My focus was on the host vehicle movement compensation, design of the multiple model filter and the evaluations.

Paper III [3]: Clothoid-based road geometry estimation using moving vehicles and road-side observations
Paper III is an extension of Paper II. The two papers are different from the following three aspects. First, besides the lane marking measurements and the movement of the leading vehicles, we also use measurements from

the guard rails to update the road state. In this paper we fit third degree polynomials to the measurements received from the left and the right guard rails using a Hough transform. Based on that we form hypotheses about the shape of the road given that normally the guard rails are parallel to the road. In each hypothesis, a subset of the received measurements is considered to have been originated form one of the guardrails and the rest are clutter, i.e., they are originated from off-road objects (inside or outside of the road). We present probabilistic models for the cardinality of the measurements received from each guard rail as well as the clutter measurements. Moreover, we develop the likelihood function of each type of measurement. Second, we present a deeper analysis of our road model. Third, we extend our evaluations to more winding highways.

Much of the work was carried out in close collaboration with the co-authors. My focus has been on the design of the cardinality models, the likelihood functions and the derivations of the posterior density resulting from the update by stationary detections.

Paper IV [4]: Variational Bayesian EM for SLAM
In this paper, a radar-based simultaneous localization and mapping (SLAM) algorithm using variational Bayesian expectation maximization (VBEM) is presented. This work is an extension of the efficient mapping algorithm presented in [14]. The joint distribution of the map, sensor poses and data association variables is computationally complex and we need approximations to handle it. The VBEM transforms the complicated inference algorithm to an optimization one and allows us to derive closed form solutions for parameters of the map, sensor poses and data associations. In this algorithm the joint posterior density of the map, the poses and the data associations are approximated as a product of three variational distributions. These distributions are found using the VBEM method. From the estimated joint posterior density, the parameters of the map, the data association probabilities and the poses are inferred.

I carried out the derivations and implementations in collaboration with the co-authors.

Paper V [5]: Poisson multi-Bernoulli conjugate prior for estimation of both detected and undetected extended objects
In this paper we show that the Poisson multi-Bernoulli mixture (PMBM) process is a conjugate prior for the inhomogeneous Poisson measurement model for extended objects. This conjugacy results in the exact derivation of a multi-object filter for extended objects. The PMBM is a conjugate prior that does not require labels to be attached to the elements of the

underlying random finite sets. In addition, this type of modeling is shown to have many advantages for tracking point objects [15], and our method carries over these advantages to tracking of extended objects. For example, the PMBM model provides an elegant way of describing objects that have been detected and those which have not been detected so far, at each time step. The exact form of the filter is too computationally complex, therefore we implement an approximated version of the filter and demonstrate that it outperforms the CPHD filter in certain scenarios.

I participated in the discussions and collaborated in the derivations with the co-authors.

Paper VI [6]: Poisson Multi-Bernoulli Radar Mapping Using Gibbs Sampling

Sensor maps help a self-driving car to localize itself. In order to have an accurate localization, the car requires detailed sensor maps. Radar is a robust and cost-effective sensor which is already mounted in some smart vehicles to perform driver assistance tasks. As such, radars are suitable candidates to be included in an accurate localization system.

In this paper we develop a radar mapping algorithm assuming that the map has a Poisson process prior. This is a new type of radar map that allows us to incorporate many different aspects into one model, namely, the uncertainties in the number of present landmarks, uncertainties in the position and extent of the landmarks, unknown data associations and description of explored and unexplored areas of the map through modeling the detected and undetected landmarks. We perform a batch update on the Poisson prior that results in a Poisson multi-Bernoulli mixture posterior. We derive the exact batch multi-object posterior density using the results in Paper V. In addition, we marginalize out the continuous states of the map and use Gibbs sampling to sample from the measurement partitions. An estimate of the map is provided by averaging over samples of the MCMC chain, avoiding the samples of the burn-in period. Moreover, we show that our method is parallelizable making it suitable for large scale problems.

I carried out the derivations and implementations in collaboration with the co-authors.

Paper VII [7]: Poisson multi-Bernoulli filter for extended object tracking

In this paper, we develop a track-oriented algorithm for extended object tracking based on the PMBM conjugate prior. This filter recursively approximates the PMBM of the posterior with a Poisson multi-Bernoulli distribution which is simpler and less computationally complex than the PMBM.

The relation between the PMB and the PMBM is in many ways similar to how the LMB filters serve as a computationally efficient approximation to the GLMB distribution. An important difference is that the LMB filter is based on an LMB birth while the PMB incorporates an unlabeled Poisson birth process. The PMB filter is even more closely connected to the track-oriented marginal MeMBer-Poisson filter (TOMB/P) [16] for point objects. An important difference between the PMB for extended objects and the TOMB/P for point objects, is that it is much more challenging to create new tracks in a reasonable manner for extended targets. We utilize a combination of pre-clustering, recycling [15] and an equivalence class among PMBM distributions in order to create a small set of new tracks at each time step. In addition, we use Gibbs sampling [17] to approximate the marginal distributions of the different tracks.

I carried out the derivations and implementations in collaboration with the co-authors.

1.3 Future Work

Following the methods and ideas presented in the accompanying papers, a whole host of new ideas can be envisioned as future extensions. In this Section, some of these ideas are discussed in detail and some are proposed as possible future threads to investigate.

- **Improvements to the mapping algorithm**
  The presented work in Paper VI [6] could be further improved by using a split-merge MCMC method similar to the one described in [18]. Our method samples from the measurement partitions by moving one measurement between the cells per iteration; the method in [18] could potentially sample more effectively by moving groups of measurements to explore the space of all possible partitions of the measurements.

  In our work we have assumed that the measurement noise of the sensor is negligible compared to the extent of the landmarks. Taking the sensor's measurement noise into account introduces non-linearities. In addition, we have used a simple gamma reduction technique for evaluating the weight of each hypothesis. An interesting line of research would be to develop mapping algorithms using the sampling method of [18], taking the sensor's measurement noise into account, including a nonlinear measurement model, and/or using a more accurate gamma mixture reduction technique.

- **Dynamic maps**
  A natural extension to the mapping method described in Paper VI [6]

is to incorporate the appearance and disappearance of landmarks. Such a mapping algorithm could be used when for example the car is revisiting an area after sometime and has an old map of that area which might have changed since the last visit.

- **Sampling trajectories**
  Sampling the measurement partitions in Paper VI [6] can be extended to sequential methods where, in effect, it would result in sampling trajectories. A prerequisite for that is to derive the multi-trajectory posterior density, an example of which can be found in [19].

- **Joint dynamic and static multi-object tracking**
  Papers V [5] and VI [6] present multi-object mapping and multi-object tracking, respectively. These two solutions could benefit from each other. That is, tracking of the moving objects could benefit from considering the updated uncertainties of the map, and the mapping algorithm could benefit from having knowledge about the position of the moving objects and their uncertainties. A joint multi-object estimation method can be devised using RFS-based models.

- **SLAM**
  In Paper VI [6] we present a mapping method, i.e., we assume that we have accurate knowledge about the pose of the sensor at each time, such that the uncertainties in the pose can be neglected. An extension could be to relax this assumption and, for example, use a particle MCMC (PMCMC) method to solve a simultaneous localization and mapping (SLAM) problem where the prior for the map is a Poisson process.

- **3-D Road Model**
  The road model that is presented in Papers II [2] and III [3] assumes that the road has a planar shape. Essentially, this assumption never holds and is particularly bad for hilly roads; however, it provides us with good approximations in many scenarios. Incorporating the slope of the road into the road model should result in a more accurate and robust model, especially for hilly roads.

- **More comprehensive road model**
  The current road model considers only one road, i.e., aspects such as exits, forks, intersections, etc. are not accounted for. Extending the road model such that it includes these aspects could be an interesting topic for future studies. For example, such a model can be used to

notify the driver if it is time to perform a lane change necessary for reaching an exit in time.

- **Use of Detailed Digital Maps and GPS**
  The sensor setup in paper II [2] and paper III [3] includes camera and radar besides the inertial measurement sensors and the wheel speed sensors. A possible future extension is to investigate how information from GPS and detailed digital maps can be incorporated into the current sensor fusion algorithm.

- **Performance improvement using unfiltered detections**
  The measurements with which we update the road state in Papers II [2] and III [3] are filtered states delivered by the sensors, i.e., we are provided with the means of some distributions and we have no information regarding their associated covariances. Consequently, possible correlations between these filtered states are not accounted for. An open question is how much the performance would improve if unfiltered detections were available.

1.4 Thesis Outline

The thesis is divided into two parts. The first provides an overview of the theoretical concepts used in the appended papers and a survey of current systems and tools. In this part, Chapter 2 provides an overview to probabilistic models and inference techniques which are used in Papers IV [4], VI [6] and VII [7]. Nonlinear filtering and mapping/tracking multiple extended objects are discussed in Chapter 3 and 4, respectively. Chapter 5 presents the concept of road geometry estimation, how nonlinear filtering can be applied to it and a brief survey of the existing algorithms. Part II includes the appended papers.

Chapter 2

Probabilistic Models and Inference Techniques

2.1 Introduction

In many applications the aim is to answer some queries regarding a phenomenon based on a probabilistic model of that phenomenon. For example, given a map of an environment and sensor observations of the relative position of a robot to some of the objects in that environment, a relevant query is: what is the most probable location of the robot in that environment?

Statistical modeling provides a probabilistic description of a phenomenon using observations of this phenomenon. We use probabilistic descriptions (models) to encode our uncertainties regarding a phenomenon. The uncertainties can arise due to our limited understanding of the phenomenon, limitations we face in the modeling of the phenomenon or the phenomenon being random in nature [20]. Statistical inference allows us to answer different queries based on a probabilistic description of a phenomenon.

Bayesian statistics is one of the tools that facilitates statistical inference by enabling us to express posterior distributions of quantities of interest. Let $\mathbf{z}$ denote observations of an event and $\mathbf{x}$ denote the unknown parameters of a probabilistic model of this event. A Bayesian statistical model includes a prior distribution of the parameters of the model $p(\mathbf{x})$ and a measurement model $p(\mathbf{z}|\mathbf{x})$. The prior summarizes our knowledge regarding the parameters $\mathbf{x}$ before making any observations. The measurement model describes the relation between the observations (measurements) and the parameters. Applying Bayes rule, the result is the posterior density of the parameters given the observations,

$$p(\mathbf{x}|\mathbf{z}) = \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{\int p(\mathbf{z}|\mathbf{x})p(\mathbf{x})d\mathbf{x}}. \tag{2.1}$$

In Bayesian methods, inference is based on the posterior density. For example, one could calculate the moments of the posterior density by computing the corresponding integrals.

For many models of interest, the main challenge is that the posterior density is so complex that performing inference on it is intractable. For example, calculating the moments of a posterior density might be intractable because the corresponding integrals do not have a closed form solution. Moreover, we are often interested in marginal distributions of some (or one) of the underlying variables which involves integrating out high dimensional variables. In case of having discrete random variables summing out all configurations of some of the variables might be a formidable task. For these reasons, we often resort to performing approximations. Two common strategies to approximate the posterior density are Markov chain Monte Carlo methods and variational methods. Both of these enable us to conveniently perform inference and compute marginal densities even if the true posterior density is very complex

In many practical systems, the underlying probabilistic model includes a large number of variables. For such complex systems, specifying a joint posterior density over all the underlying variables might be a daunting task. Probabilistic graphical models (PGM) are tools with which we can represent complicated models involving large number of variables. This type of representation allows us to visualize the model and gain insights into the properties of a highly complex model [21]. PGMs are compact representations that encode the conditional independencies among the variables of a joint probability distribution, thus revealing the structure of the distribution and how it can be factorized [20]. We often try to leverage on conditional independences in order to devise scalable and accurate algorithms and the PGMs play an important role in the design of such algorithms [22]. In fact, the PGMs can be viewed as a suitable language to describe these algorithms.

The focus of this chapter is PGMs, Gibbs sampling [21, 23, 24] and Variational Bayesian expectation maximization (VBEM) [25] as two methods by which we can approximate posterior densities of interest.

2.2 Probabilistic graphical models

A probabilistic graphical model represents a joint probability density over the underlying variables of a probabilistic model. In a PGM, each variable (or group of variables) is represented by a node. An edge (also referred to as an arc) between two nodes describes the probabilistic interaction between the nodes. There are two types of edges: directed and undirected. Directed edges imply causality while undirected edges describe an interaction of a

different nature. For example undirected edges can be interpreted as agreement or compatibility. Two major classes of PGMs are Bayesian networks which are directed graphical models, and Markov random fields which are undirected graphs. In some inference applications it is convenient to use a different representation called a factor graph [26], which is a different type of undirected graph.

A Bayesian network is a directed acyclic graph whose nodes represent random variables [20]. A directed acyclic graph is a graph where one can not end up on the same starting node by following the directed edges of the network [21]. For example, consider the Bayesian network illustrated in Figure 2.1. The joint probability density that factorizes according to this graph is given by

$$p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) = p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{x}_4)p(\mathbf{x}_3|\mathbf{x}_1)p(\mathbf{x}_1)p(\mathbf{x}_4), \tag{2.2}$$

where the random variables of the model are denoted by $\mathbf{x}_i$.

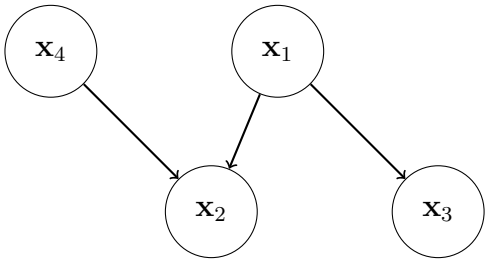

Figure 2.1: An example of a Bayesian network. The joint density expressed in (2.2) satisfies all the conditional independencies encoded in this Bayesian network.

Markov random fields are undirected graphs where the nodes represent random variables, or sets of variables, and the edges imply probabilistic interaction (soft constraints between the connecting nodes [21]). This type of model is useful in situations where it is difficult to formulate causality between the random variables. In the context of Markov random fields, we deal with factors of random variables rather than conditional probability distributions. A factor of some variables is a function of these variables and does not have to sum to one (does not have to be a probability density). Let us consider the Markov random field depicted in Figure 2.2. One joint probability distribution that factorizes according to this graph is expressed as

$$p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) = \frac{1}{Z}\phi(\mathbf{x}_1, \mathbf{x}_2)\phi(\mathbf{x}_1, \mathbf{x}_3)\phi(\mathbf{x}_2, \mathbf{x}_3)\phi(\mathbf{x}_2, \mathbf{x}_4)\varphi(\mathbf{x}_3, \mathbf{x}_4), \tag{2.3}$$

where $\mathbf{x}_i$s denote the random variables of the model, $\phi(\mathbf{x}_i, \mathbf{x}_j)$ is a factor describing the interaction between $\mathbf{x}_i$ and $\mathbf{x}_j$, and Z is a normalizing constant that is computed by integrating (summing) the product of the factors over all the variables. In general, a distribution is said to factorize according to a Markov random field whenever there is an edge between all pairs of variables that appear in the same factor.

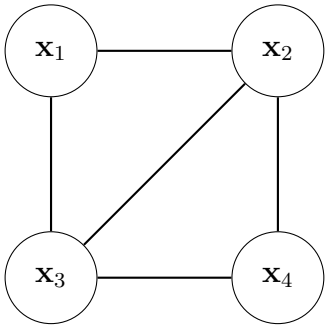

Figure 2.2: An example of a Markov random filed. The joint density expressed in (2.3) factorizes according to this graph.

Both Bayesian networks and Markov random fields encode conditional independencies and are related to techniques to express a joint probability density over a set of variables as a product of factors over subsets of these variables [21]. Factor graphs is an alternative representation that reveals the structure explicitly by introducing additional nodes for the factors [20]. That is, a factor graph has two kinds of nodes: a variable node for each of the variables of the joint density, and a factor node that describes the interaction among the variables it is connected to. A factor graph is an undirected graph that only contains edges between the variable nodes and factor nodes [20]. Note that the Markov random field in Figure 2.2 does not specify whether the underlying factors are the pairwise factors of (2.3) or if the factors are, e.g., $\phi(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ and $\phi(\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$. By contrast the factor graph of Figure 2.3 corresponding to (2.3) depicts the explicit structure of the factors.

2.3 Gibbs sampling

Gibbs sampling [23] is a simple and popular type of Markov chain Monte Carlo algorithm. Each step of this method involves drawing a sample from the conditional distribution of one of the variables given the rest of the variables. For instance, let $p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ be the distribution we wish to

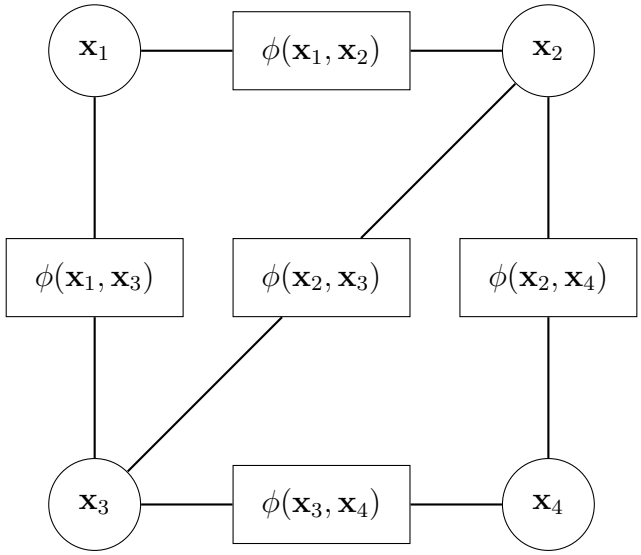

Figure 2.3: A factor graph representation of the joint density expressed in (2.3).

sample froms, and assume that at step i of the algorithm we have the following joint state $\mathbf{X}^i = [\mathbf{x}_1^i, \mathbf{x}_2^i, \mathbf{x}_3^i]$. The Gibbs sampling procedure that generates $\mathbf{X}^{i+1}$ can be summarized by

$$\begin{array}{ccc} \mathbf{x}_1^{i+1} \sim p(\mathbf{x}_1|\mathbf{x}_2^i, \mathbf{x}_3^i) & \mathbf{x}_2^{i+1} = \mathbf{x}_2^i & \mathbf{x}_3^{i+1} = \mathbf{x}_3^i \\ \mathbf{x}_1^{i+2} = \mathbf{x}_1^{i+1} & \mathbf{x}_2^{i+2} \sim p(\mathbf{x}_2|\mathbf{x}_1^{i+1}, \mathbf{x}_3^{i+1}) & \mathbf{x}_3^{i+2} = \mathbf{x}_3^{i+1} \\ \mathbf{x}_1^{i+3} = \mathbf{x}_1^{i+2} & \mathbf{x}_2^{i+3} = \mathbf{x}_2^{i+2} & \mathbf{x}_3^{i+3} \sim p(\mathbf{x}_3|\mathbf{x}_1^{i+2}, \mathbf{x}_2^{i+2}). \end{array} \tag{2.4}$$

In general, at each step of the algorithm we sample from the full conditional density of the nth variable [27], i.e., $p(\mathbf{x}_n|\mathbf{X}_{-n})$, where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_N]$ and $\mathbf{X}_{-n}$ contains all the variables except $\mathbf{x}_n$. Given large number of samples and under reasonably general conditions, the distribution of $\mathbf{X}^i$ converges to $p(\mathbf{X})$ [24] [1]. Therefore, the Gibbs sampling method provides an efficient alternative to sampling a high dimensional joint probability density by drawing samples from lower dimensional conditional distributions. Accordingly, the applicability of this method depends on how easy it is to sample from the conditional distributions.

As an example, consider the joint Gaussian distribution $p(x_1, x_2) = \mathcal{N}([x_1, x_2]; \mu, \mathbf{\Sigma})$, where $\mu = [1, 3]$ and $\mathbf{\Sigma} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix}$. To generate

[1] For notational convenience, we use $p(\mathbf{X})$ to denote the posterior density in the rest of this chapter

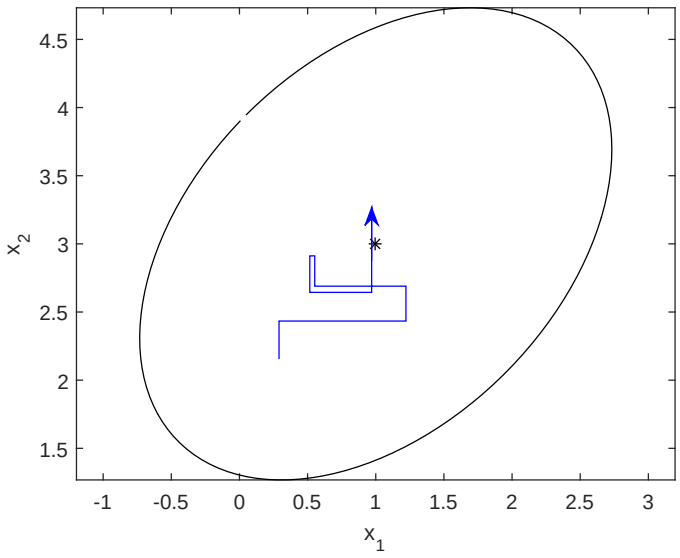

Figure 2.4: Illustration of the Gibbs sampling procedure of alternately sampling from x_1 and x_2 together with 3-σ ellipse of the joint density.

samples from this distribution, we run a Gibbs sampler that samples form $p(x_1|x_2)$ and $p(x_2|x_1)$. Figure 2.4 illustrates the first few samples generated by the Gibbs sampler. As we can see, the sampler is moving in the domain of the joint density. After running the sampler for some time, enough time for the sampler to reach its stationary distribution, we expect the samples to be from the joint density. Figure 2.5 depicts 500 final samples of the MCMC chain together with 3-σ ellipse of the joint density. The chain is run for 5000 iterations. As we can see the samples fit the joint density reasonably well.

In some models we can analytically integrate out part of the underlying variables and sample the rest. Therefore, only the remaining variables (those which are not integrated out) participate in the Markov chain. If we run a Gibbs sampler on the remaining variables, we obtain an algorithm referred to as collapsed Gibbs sampling [27]. This sampler tends to be more efficient than the basic Gibbs sampler, since we are sampling in a lower dimensional space. In Papers VI [6] and VII [7], we integrate out the continuous states and sample from the discrete states, using collapsed Gibbs sampling.

2.4 Variational Bayesian expectation maximization

Variational methods are another approach by which we can approximate the posterior density. The idea is to approximate the posterior $p(\mathbf{X})$ by a

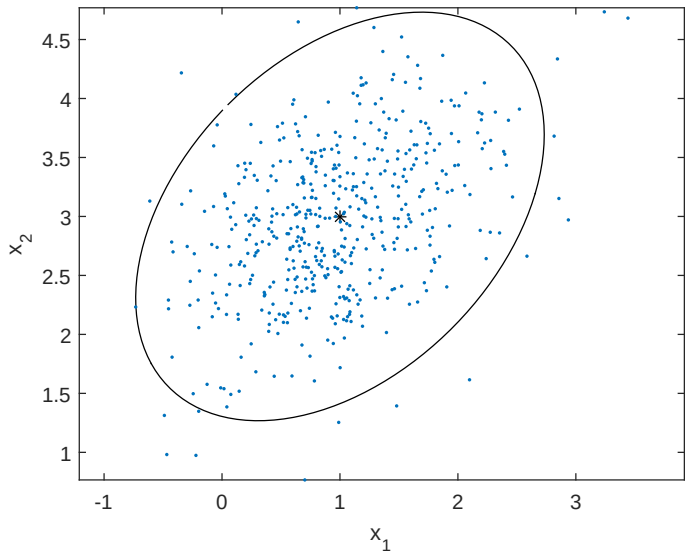

Figure 2.5: The 500 final samples of the Gibbs sampler and the 3-σ region of the joint density.

simpler probability distribution $q(\mathbf{X})$ that is tractable [25].

In the Variational Bayesian expectation maximization (VBEM) framework, we are interested in approximating an intractable joint probability density over possibly large number of variables. The approximation is based on assuming certain independencies between some of the variables. suppose that we have separated the variables in $\mathbf{X}$ into two groups $\mathbf{X}_1$ and $\mathbf{X}_2$. The basic idea in the VBEM is to approximate the posterior by

$$p(\mathbf{X}) \approx q_1(\mathbf{X}_1)q_2(\mathbf{X}_2), \tag{2.5}$$

such that $q_1(\mathbf{X}_1)q_2(\mathbf{X}_2)$ is close to the posterior in the "exclusive" Kullback-Liebler (KL) divergence sense [28]. This approximation serves the purpose of obtaining a tractable algorithm. Note that by approximating the posterior by the product in (2.5), we assume that each group of the variables is independent from the other group.

Consequently, the problem of approximating the posterior density is translated into an optimization problem where the aim is to minimize the KL divergence expressed by

$$\mathbb{KL}(q_1(\mathbf{X}_1)q_2(\mathbf{X}_2)||p(\mathbf{X})) = \int\int q_1(\mathbf{X}_1)q_2(\mathbf{X}_2)\log\frac{q_1(\mathbf{X}_1)q_2(\mathbf{X}_2)}{p(\mathbf{X})}d\mathbf{X}_1 d\mathbf{X}_2. \tag{2.6}$$

The VBEM minimizes the KL divergence in (2.6) by alternately fixing the distribution over one set of variables and minimizing the KL divergence

with respect to the other set. The VBEM is an iterative algorithm where each iteration consists of the following calculations

$$\begin{aligned} q_1^{i+1}(\mathbf{X}_1) &\propto \exp\left(\int q_2^i(\mathbf{X}_2)\log p(\mathbf{X})d\mathbf{X}_2\right) \\ &= \exp\left(\mathbb{E}_{q_2^i}\{\log p(\mathbf{X})\}\right) \qquad (2.7) \\ q_2^{i+1}(\mathbf{X}_2) &\propto \exp\left(\int q_1^{i+1}(\mathbf{X}_1)\log p(\mathbf{X})d\mathbf{X}_1\right) \\ &= \exp\left(\mathbb{E}_{q_1^{i+1}}\{\log p(\mathbf{X})\}\right), \qquad (2.8) \end{aligned}$$

where $\mathbb{E}_{q_j}\{\cdot\}$ denotes the expected value w.r.t. the distribution $q_j(\cdot)$. For models in the conjugate-exponential family the variational distributions in (2.7) and (2.8) have closed form solutions [29]. This family includes models where the data likelihood belongs to the exponential family and the prior is conjugate to that [29].

Using VBEM, we trade accuracy for speed. The VBEM is guaranteed to monotonically decrease the KL divergence (increase a lower bound on the log marginal likelihood) [25]. However, this does not guarantee global convergence. In fact, since the VBEM is only guaranteed to converge to a local optimum, the convergence point might not be a favorable point and this is one of the drawbacks of the VBEM. Another disadvantage is that this method often results in an approximated posterior that is overconfident. This is due to the use of the "exclusive" KL divergence [27]. The benefit of the VBEM algorithm is that it is computationally efficient. In Paper IV [4], we use VBEM to solve a simultaneous localization and mapping (SLAM) problem.

Chapter 3

Nonlinear Filtering

3.1 Problem Formulation

The objective in filtering is the sequential estimation of some dynamic parameters/characteristics of interest (of a system) given noisy observations from it. The parameters of interest, denoted by $\mathbf{x}_k \in \mathbb{R}^{n_x}$, describe the state of a system at time k and often we are not able to make a direct observation of them. The sequence of observations made on the system up to and including time k is denoted by $\mathbf{z}_{1:k}$.

To analyze a dynamic system from a Bayesian point of view, at least two models are required [30]. One that describes the time evolution of the state vector (process model) and another that establishes the relationship between the state vector and the noisy observations (measurement model). Generally, these two models can be time-variant, however for notational convenience we assume them to be time-invariant. Moreover, we assume that the underlying uncertainties can be modeled by additive noise. Using these assumptions, our state-space model is expressed as

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1} \tag{3.1}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k \tag{3.2}$$

where $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are the motion model function and the measurement model function, respectively. The underlying uncertainties of the process model is accounted for by the process noise $\mathbf{v}_{k-1}$ with covariance $\mathbf{Q}$. The measurement noise and its covariance are denoted by $\mathbf{w}_k$ and $\mathbf{R}$, respectively. Additionally, $\mathbf{v}_{k-1}$ and $\mathbf{w}_k$ are generated by independent white processes.

In this thesis, (3.1) is referred to by the term *process (motion) model*, this model can also be described by a conditional probability density function (pdf), $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. Similarly, (3.2) can be expressed as a conditional pdf, $p(\mathbf{z}_k|\mathbf{x}_k)$, and is referred to by the term *measurement model*.

3.2 Optimal Bayesian Solution

In the Bayesian approach we attempt to recursively calculate the posterior density of the state given all the observations up to and including time k, $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. In this context, the solution has two steps, prediction and measurement update. In the prediction step we use (3.1) to align the time index of the state vector with the received measurements. More specifically, assuming that we know $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ from the previous time step, we can use the Chapman-Kolmogorov equation to calculate the predicted density,

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_{1:k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}. \quad (3.3)$$

The predicted state is typically estimated by the mean of this density which is calculated by

$$\hat{\mathbf{x}}_{k|k-1} = \int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{z}_{1:k-1})d\mathbf{x}_k. \quad (3.4)$$

In the measurement update we update the predicted density by conditioning on the measurement received at time k using (3.2) and Bayes' rule

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}_{1:k-1})p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \quad (3.5)$$

where

$$p(\mathbf{z}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}_{1:k-1})p(\mathbf{x}_k|\mathbf{z}_{1:k-1})d\mathbf{x}_k. \quad (3.6)$$

The posterior density summarizes all the available information about the state at time k. Using this density we can calculate state estimates according to different optimality criteria. For example, the minimum mean squared error (MMSE) estimate minimizes the mean squared error and is calculated by

$$\hat{\mathbf{x}}_k^{\text{MMSE}} = \arg\min_{\hat{\mathbf{x}}_k} \text{tr}(\mathbb{E}\{(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T\}), \quad (3.7)$$

where $\text{tr}(\cdot)$ is the the trace operator, $\mathbb{E}\{\cdot\}$ denotes the expected value which is calculated w.r.t the posterior density. The solution to (3.7) is the conditional mean of the posterior density and is expressed as

$$\hat{\mathbf{x}}_k^{\text{MMSE}} = \int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{z}_{1:k})d\mathbf{x}_k. \quad (3.8)$$

Another example is the maximum a posteriori (MAP) estimate which is given by

$$\hat{\mathbf{x}}_k^{\text{MAP}} = \arg\max_{\mathbf{x}_k} p(\mathbf{x}_k|\mathbf{z}_{1:k}). \quad (3.9)$$

The optimal Bayesian solution is not always achievable. In many cases the solutions to (3.3), (3.5) and (3.8) are intractable. A special case when an optimal solution can be achieved if both process and measurement models are linear and their respective noises are additive, white and Gaussian. In this situation the solution to (3.8) and (3.9) are equal and the Kalman filter provides the optimal solution.

In cases where our problem does not satisfy the linearity or the Gaussian conditions, finding an optimal solution is often very complicated if not impossible. Consequently, in most cases, we have to give up optimality and settle for a suboptimal solution, a large number of which is already available in the literature. For example, we can find many suboptimal solutions in the family of linear minimum mean squared error (LMMSE) filters [30,31]. Additionally, suboptimal MAP estimators can be found in iterative Gaussian filters [32, 33]. Particle filters are another alternative solution [30].

In this thesis we mainly focus on the first two family of filters since we are interested in low-complexity algorithms that can provide us with an accurate solution, such that they can be implemented in real time with moderate computational resources.

3.3 LMMSE Filters

The family of linear minimum mean squared error (LMMSE) filters yields the state estimate as a linear function of the measurements. This estimate is the best linear estimate in the MMSE sense. The LMMSE estimate is the optimal Bayesian solution, in both MMSE and MAP sense, when the models are linear and Gaussian. Given a measurement $\mathbf{z}$, the LMMSE estimate of the state $\mathbf{x}$ is expressed by

$$\hat{\mathbf{x}}^{\text{LMMSE}} = \bar{x} + \mathbf{P}_{xz}\mathbf{P}_{zz}^{-1}(\mathbf{z} - \bar{\mathbf{z}}), \tag{3.10}$$

where $\bar{x}$ and $\bar{\mathbf{z}}$ are the mean of $\mathbf{x}$ and $\mathbf{z}$, respectively. In addition, $\mathbf{P}_{xz}$ is the cross covariance of $\mathbf{x}$ and $\mathbf{z}$, and $\mathbf{P}_{zz}$ is the covariance of the measurement.

The connection between the LMMSE estimation and the Bayesian filtering problem can be established by examining the prediction and update steps for both a linear Gaussian setting and a nonlinear setting. That is, the problem of finding the predicted state can be viewed as an estimation problem where we seek to find the LMMSE estimate of $\mathbf{x}_k$ given $\mathbf{z}_{1:k-1}$. Similarly, calculating the updated state can be viewed as solving an estimation problem where we are interested in finding the LMMSE estimate of $\mathbf{x}_k$ given $\mathbf{z}_k$.

When the process model is linear and Gaussian, the predicted density

is Gaussian and is given by

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}). \tag{3.11}$$

The LMMSE estimate of $\mathbf{x}_k$ given $\mathbf{z}_{1:k-1}$, in this particular situation, happens to be the conditional mean of the predicted density, and can be calculated analytically. If the measurement model is Gaussian, the posterior density is Gaussian as well and is expressed by

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}). \tag{3.12}$$

In this case, the LMMSE estimate of $\mathbf{x}_k$ given $\mathbf{z}_{1:k}$ is the conditional mean of the posterior density, i.e., the solution to the integral in (3.8). In the linear Gaussian case, this integral has a closed form solution which is linear in $\mathbf{z}_k$.

In the case of having a nonlinear process model, the predicted density is not Gaussian, and, the solution to (3.4) is not linear in $\mathbf{z}_{1:k-1}$. Therefore, the prediction step can no longer be viewed as an LMMSE estimator. A nonlinear measurement model results in a non-Gaussian posterior density. As a result, the solution to (3.8) often does not have a closed form. Therefore, it seems that the updated state is no longer an LMMSE estimate. However, if we seek to approximate the posterior density by a Gaussian distribution, it is sufficient to estimate its first two moments and one possible way to approximate the mean of the posterior is LMMSE estimation. In fact, many of the most popular Gaussian filters use LMMSE estimation in the update step.

3.3.1 Kalman Filter

Kalman filter [34] is the recursion of (3.3) and (3.5) in the linear Gaussian case. Under these assumptions, the Kalman filter provides us with the optimal Bayesian solution which can be calculated by

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1} \\
\mathbf{P}_{k|k-1} &= \mathbf{Q} + \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T \qquad (3.13) \\
\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) \\
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \qquad (3.14)
\end{aligned}$$

where $\mathbf{F}$ is a known matrix of the linear process model, $\mathbf{K}_k$ is the Kalman gain and $\mathbf{S}_k$ is the covariance of the innovation $(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1})$ which is calculated by

$$\mathbf{S}_k = \mathbf{R} + \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T. \tag{3.15}$$

Additionally, $\mathbf{K}_k$ is the Kalman gain and is expressed as

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}_k^{-1}, \tag{3.16}$$

where $\mathbf{P}_{k|k-1}\mathbf{H}^T$ is the cross covariance between $\mathbf{x}_k$ and $\mathbf{z}_k$, and $\mathbf{S}_k$ is the covariance of $\mathbf{z}_k$ given $\mathbf{z}_{1:k-1}$. Therefore, we can see that the Kalman gain is equivalent to the matrix that appears in (3.10).

3.3.2 Extended Kalman Filter

The extended Kalman filter (EKF) [30] approximates the nonlinear process and measurement models as affine using the first order Taylor series expansion of the nonlinear models.

In the prediction step, $\mathbf{f}(\cdot)$ is approximated by its first order Taylor series expansion about $\hat{\mathbf{x}}_{k-1|k-1}$. Consequently, the predicted state and its covariance are given by

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}) \\ \mathbf{P}_{k|k-1} &= \mathbf{Q} + \hat{\mathbf{F}}\mathbf{P}_{k-1|k-1}\hat{\mathbf{F}}^T \end{aligned} \tag{3.17}$$

where $\hat{\mathbf{F}}$ is the Jacobian of $\mathbf{f}(\cdot)$, i.e.,

$$\hat{\mathbf{F}} = [\nabla_{\mathbf{x}_{k-1}}\mathbf{f}^T(\mathbf{x}_{k-1})]^T|_{\mathbf{x}_{k-1}=\hat{\mathbf{x}}_{k-1|k-1}}$$

where

$$\nabla_{\mathbf{x}_{k-1}} = \left(\frac{\partial}{\partial(\mathbf{x}_{k-1}(1))} \quad \frac{\partial}{\partial(\mathbf{x}_{k-1}(2))} \quad \cdots \quad \frac{\partial}{\partial(\mathbf{x}_{k-1}(n_x))} \right)^T. \tag{3.18}$$

Accordingly, the measurement model is approximated by its first order Taylor series expansion about $\hat{\mathbf{x}}_{k|k-1}$ and the update step can be expressed as

$$\begin{aligned} \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})) \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \end{aligned} \tag{3.19}$$

where

$$\begin{aligned} \mathbf{S}_k &= \mathbf{R} + \hat{\mathbf{H}}\mathbf{P}_{k|k-1}\hat{\mathbf{H}}^T \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}\hat{\mathbf{H}}^T\mathbf{S}_k^{-1} \end{aligned} \tag{3.20}$$

and

$$\hat{\mathbf{H}} = [\nabla_{\mathbf{x}_k}\mathbf{h}^T(\mathbf{x}_k)]^T|_{\mathbf{x}_k=\hat{\mathbf{x}}_{k|k-1}}$$

is the Jacobian of $\mathbf{h}(\cdot)$.

It should be noted that in order to use the EKF we need the Jacobian of the nonlinear models. In the cases where the Jacobian can not be readily calculated, we need to turn to suboptimal methods where we do not use the Jacobian directly. Moreover, in situations where the first order Taylor series expansion does not provide a good description of the nonlinear models, we need to look for better suboptimal methods.

3.3.3 Unscented Kalman Filter

The unscented Kalman filter (UKF) [35] uses the unscented transform in the prediction and update step. The unscented transform is a method to approximate the first two moments of a random variable that has been propagated through a nonlinear function. In other words, this method calculates the first two moments of the underlying densities by numerical integration.

Analogous to our filtering problem, to calculate the integral in (3.3), we represent $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ by a set of N deterministic samples $\mathbf{x}_{k-1}^i$ also referred to as sigma points, and their weights W_{k-1}^i. The sigma points and their weights are chosen such that they completely capture the first two moments of the distribution, i.e.,

$$\begin{aligned}
\mathbf{x}_{k-1}^0 &= \hat{\mathbf{x}}_{k-1|k-1} & W_{k-1}^0 &= \frac{\kappa}{\kappa + n_x} \\
\mathbf{x}_{k-1}^i &= \hat{\mathbf{x}}_{k-1|k-1} + \left(\sqrt{(n_x+\kappa)\mathbf{P}_{k-1|k-1}}\right)_i & W_{k-1}^i &= \frac{1}{2(n_x+\kappa)} \\
\mathbf{x}_{k-1}^{i+n_x} &= \hat{\mathbf{x}}_{k-1|k-1} - \left(\sqrt{(n_x+\kappa)\mathbf{P}_{k-1|k-1}}\right)_{i+n_x} & W_{k-1}^{i+n_x} &= \frac{1}{2(n_x+\kappa)}
\end{aligned} \tag{3.21}$$

where $i = 1, 2, ..., n_x$ and κ is a constant called the scaling factor which can be viewed as a design parameter. Additionally, $\left(\sqrt{(n_x+\kappa)\mathbf{P}_{k-1|k-1}}\right)_i$ is the i^{th} row of $\mathbf{L}$ such that

$$(n_x+\kappa)\mathbf{P}_{k-1|k-1} = \mathbf{L}^T\mathbf{L}. \tag{3.22}$$

Accordingly, the prediction step is expressed as

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k-1} &= \sum_{i=0}^{N-1} W_{k-1}^i \mathbf{f}(\mathbf{x}_{k-1}^i) \\
\mathbf{P}_{k|k-1} &= \mathbf{Q} + \sum_{i=0}^{N-1} W_{k-1}^i (\mathbf{x}_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1})^T
\end{aligned} \tag{3.23}$$

where $\mathbf{x}_{k|k-1}^i = \mathbf{f}(\mathbf{x}_{k-1}^i)$ are sigma points of the predicted density. The update step follows the equations in (3.14) but with

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{P}_{xz}\mathbf{S}_k^{-1} \\
\mathbf{S}_k &= \mathbf{R} + \mathbf{P}_{zz} \\
\mathbf{P}_{xz} &= \sum_{i=0}^{N-1} W_{k-1}^i (\mathbf{f}(\mathbf{x}_{k-1}^i) - \hat{\mathbf{x}}_{k|k-1})(\mathbf{h}(\mathbf{x}_{k|k-1}^i) - \hat{\mathbf{z}}_{k|k-1})^T \\
\mathbf{P}_{zz} &= \mathbf{R} + \sum_{i=0}^{N-1} W_{k-1}^i (\mathbf{h}(\mathbf{x}_{k|k-1}^i) - \hat{\mathbf{z}}_{k|k-1})(\mathbf{h}(\mathbf{x}_{k|k-1}^i) - \hat{\mathbf{z}}_{k|k-1})^T \\
\hat{\mathbf{z}}_{k|k-1} &= \sum_{i=0}^{N-1} W_{k-1}^i \mathbf{h}(\mathbf{x}_{k|k-1}^i).
\end{aligned} \tag{3.24}$$

The update step implicitly makes use of an LMMSE estimator to select the posterior mean $\hat{\mathbf{x}}_{k|k}$.

Besides the first and second moment, the sigma points will match the skewness of the underlying density due to their symmetry [36]. Moreover, if the scaling factor is set to $\kappa = 3 - n_x$, the sigma points will capture the kurtosis of the density as well [37]. This choice can lead to a negative weight for the mean, which in turn might cause the covariances to become non-positive definite. In such an event the filter would crash as it will not be possible to calculate the spread of the sigma points, since (3.22) can not be satisfied. This indicates that for high dimensional problems, UKF is prone to numerical errors.

Square rooting in (3.22) is a numerically sensitive and a computationally expensive operation. It can be avoided by a square root implementation of UKF [35] where instead of the covariance matrices, the square root of them is propagated. However, negative weights may still give rise to problems, also in this algorithm [37].

3.3.4 Cubature Kalman Filter

The Cubature Kalman Filter (CKF) [37] is another nonlinear filtering method. Similar to UKF, the aim is to calculate the moments of the underlying densities by numerical integration. To this end, the CKF uses the properties of numerical integration methods known as the cubature rules [38]. The difference between the numerical integration methods of the UKF and the CKF results in a different set of sigma points and weights, though it is true that the cubature rule is a special case of the unscented transform.

In practice the CKF captures the first two moments of the underlying densities by $2n_x$ sigma points and their weights. The difference between the sigma points of the CKF and UKF is in one point and that is the mean of the density for which the weight is zero. For example, to capture the mean and covariance of $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$, the CKF uses the following sigma points with their associated weights,

$$\begin{aligned} \mathbf{x}_{k-1}^{i} &= \hat{\mathbf{x}}_{k-1|k-1} + \left(\sqrt{(n_x+\kappa)\mathbf{P}_{k-1|k-1}}\right)_i & W_{k-1}^{i} &= \frac{1}{2n_x} \\ \mathbf{x}_{k-1}^{i+n_x} &= \hat{\mathbf{x}}_{k-1|k-1} - \left(\sqrt{(n_x+\kappa)\mathbf{P}_{k-1|k-1}}\right)_{i+n_x} & W_{k-1}^{i+n_x} &= \frac{1}{2n_x} \end{aligned} \tag{3.25}$$

where $i = 1, 2, ..., n_x$ and it is implied that $W_{k-1}^{0} = 0$. Having the sigma points and the weights, the rest of the filtering algorithm flow is similar to the UKF.

The definition of the sigma points and their weights in (3.25) eliminates the possibility of a negative weight and results in improved numerical

robustness and availability of a robust square root implementation. This characteristic makes CKF a very interesting candidate for real-time applications.

3.4 Iterative Gaussian Filters

The iterative Gaussian filters provide us with yet another suboptimal solution to the filtering problem. To derive these filters we still rely on the assumptions we made for the LMMSE filters, i.e., the underlying densities are unimodal and can be approximated by a Gaussian distribution. Additionally, we focus on the update step of the filter.

A common type of iterative Gaussian filters are those that make use of a MAP estimate, see (3.9), to approximate the posterior mean. Given our model assumptions, (3.9) can be expressed as a the following optimization problem

$$\hat{\mathbf{x}}_k^{\text{MAP}} = \arg\min_{\mathbf{x}_k} L(\mathbf{x}) \tag{3.26}$$

where

$$\begin{aligned} L(\mathbf{x}) &= \frac{1}{2}\Bigg[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^T \mathbf{P}_{k|k-1}^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) \\ &\quad + (\mathbf{z}_k - h(\mathbf{x}_k))^T \mathbf{R}^{-1}(\mathbf{z}_k - h(\mathbf{x}_k))\Bigg] \end{aligned} \tag{3.27}$$

is the objective function, the complete derivation of which can be found in [32] and Paper I [1]. Though beyond the scope of this introduction, an interesting modern type of iterative Gaussian filter is the Iterated posterior linearization filter presented in [33].

Most strategies to solve (3.26) are iterative methods, which can be expressed as

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{c}_i, \tag{3.28}$$

where $\mathbf{c}_i$ is the correction factor whose value depends on how a descent direction and a step size over it are chosen. More specifically, the value of the correction factor depends on the optimization method. Once the optimization algorithm converges, we set $\hat{\mathbf{x}}_{k|k} = \mathbf{x}_{i+1}$ and we update the covariance. This implies that in each time step of the filter regression, an iterative optimization problem should be solved. Clearly, this solution increases the computational complexity of the filtering algorithm. In this section we explain why sometimes we are willing to pay this price.

Historically, the Gauss-Newton method [39] was the first algorithm that was applied to this problem [40]. The correction factor for this method is given by

$$\mathbf{c}_i = -(\nabla^2 L(\mathbf{x}_i))^{-1} \nabla L(\mathbf{x}_i). \tag{3.29}$$

where $\nabla^2 L(\mathbf{x}_i)$ is the Hessian of $L(\mathbf{x})$ and $\nabla L(\mathbf{x}_i)$ is its gradient. The resulting algorithm known as IEKF also IKF, depicts an interesting observation, i.e., the first iteration of this filter is equivalent to the EKF. Another observation is that in the linear Gaussian case, the solution will coincide with the Kalman filter update (MMSE solution). More specifically, in such a situation, the solution is the minimum of a quadratic convex function which can be found by performing one iteration of the Gauss-Newton method.

The convergence of the Gauss-Newton method is guaranteed if the iterations start from an initial point that is sufficiently close to the optimum [41]. In our filtering context, the initial point is often the predicted state $\hat{\mathbf{x}}_{k|k-1}$. Being close to the optimum implies that the predicted state provides us with a guess that is close to the final solution; in other words, if the measurement is very informative the convergence is not guaranteed. Therefore, another suggestion is to use the Levenberg-Marquardt method [42] to solve (3.26). In this case the correction factor is described by

$$\mathbf{c}_i = -(\nabla^2 L(\mathbf{x}_i) + \mu_i \mathbf{I})^{-1} \nabla L(\mathbf{x}_i) \tag{3.30}$$

where μ_i controls the behavior of the algorithm. The filter which is based on this method is presented in [43] and also explained by the name of LM-IEKF in Paper I [1]. This filter is more robust but also more computationally complex.

Besides these two algorithms, there are other methods that are based on the sigma-point filters. These methods do not need the Gradient and Jacobian of the measurement model. Instead the correction factor is calculated by propagating the sigma-points, see for example [32, 33, 44].

So far what we have discussed covers the calculation of the mean of the posterior density. For covariance update we have used a method in Paper I [1] that is similar to [32] and [43]. This update is similar to the covariance update of the EKF, i.e., it is performed once in each time step after the optimization algorithm has converged, and can be viewed as the correct update under the assumption that we linearize the measurement function in the update state.

The iterative Gaussian filters are preferred over the LMMSE methods when we have an informative measurement. In such a case, LMMSE methods such as EKF and UKF fail to estimate the posterior density accurately [45]. Therefore, the increased computational complexity is the price we pay for having more accuracy.

3.5 Multiple Model Filters

Multiple model filters estimate the state of a system that goes through a finite number of modes or models. For example in the context of detecting other vehicles on the road, the lateral motion of a leading vehicle can be described by two models. One that describes the following lane behavior and the other which describes the changing lane movement of the vehicle. In this problem we assume that at any given time the vehicle follows one of these two models, but we don't know which one.

In situations like this we have both continuous (Gaussian noise) and discrete (modes) uncertainty about the state of the system [31]. Such a system is called hybrid and its behavior can be described as dynamic switching model/mode jumping and a dynamic multiple model estimator is required to solve our problem.

3.5.1 Optimal Solution

A discrete-time hybrid system can be described by the following process and measurement equations

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, q_k) + \mathbf{v}_{k-1}(q_k) \tag{3.31}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, q_k) + \mathbf{w}_k(q_k) \tag{3.32}$$

where q_k is the mode variable and its time evolution follows a mode switching process. Furthermore, at time k, $q_k \in \{1, 2, ..., M\}$ and $q_k = i$ implies that i is the effective mode of the system during $(t_{k-1}, t_k]$. The switching mode process is often assumed to be a first order homogeneous Markov chain. Such a chain is statistically described by its mode transition probabilities

$$\pi_{ij} = Pr\{q_k = j | q_{k-1} = i\}$$

where

$$\sum_{j=1}^{M} \pi_{ij} = 1.$$

Additionally, (3.31) and (3.32) indicate that the structure of the system and/or the statistics of the noises can depend on the mode of the system.

We denote one sequence of the modes up to and including time k as

$$\mathbf{q}_k^l = \{q_1^l, q_2^l, ..., q_k^l\}. \tag{3.33}$$

where q_α^l for $1 < \alpha < k$ is the mode of the system at time α in the mode sequence l. Note that at time k, there are M^k possible mode sequences.

Provided that (3.33) describes a set of mutually exclusive and exhaustive events, we can use the total probability theorem to express the conditional probability of the state at time k as

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \sum_{l=1}^{M^k} p(\mathbf{x}_k|\mathbf{z}_{1:k}, \mathbf{q}_k^l) Pr\{\mathbf{q}_k^l|\mathbf{z}_{1:k}\}. \tag{3.34}$$

If all the underlying models are linear and the noises are Gaussian, the system is called jump Markov linear system (JMLS). In this case, the posterior density resulting from each mode sequence, $p(\mathbf{x}_k|\mathbf{z}_{1:k}, \mathbf{q}_k^l)$ is Gaussian. Consequently, (3.34) is a Gaussian mixture whose number of elements increases exponentially with time. To calculate each of these posterior densities we need a Kalman filter for each mode sequence. For nonlinear systems, we can approximate each posterior by a Gaussian distribution. In this situation, in order to calculate the Gaussian mixture we should run M^k parallel nonlinear filters, e.g., EKF or UKF. We denote the weight of each posterior density by $\omega_k^l = Pr\{\mathbf{q}_k^l|\mathbf{z}_{1:k}\}$ and describe it by

$$\begin{aligned} \omega_k^l &\propto p(\mathbf{z}_k|\mathbf{q}_k^l, \mathbf{z}_{1:k-1}) Pr\{\mathbf{q}_k^l|\mathbf{z}_{1:k-1}\} \\ &\propto p(\mathbf{z}_k|\mathbf{q}_k^l, \mathbf{z}_{1:k-1}) Pr\{q_k^l|q_{k-1}^l\} Pr\{\mathbf{q}_{k-1}^l|\mathbf{z}_{1:k-1}\} \\ &\propto p(\mathbf{z}_k|\mathbf{q}_k^l, \mathbf{z}_{1:k-1}) \pi_{ij} \omega_{k-1}^l \end{aligned} \tag{3.35}$$

where we have used the fact that in a Markovian process, the current state of the system only depends on its state at the previous time stamp. Moreover, $p(\mathbf{z}_k|\mathbf{q}_k^l, \mathbf{z}_{1:k-1})$ is the distribution of the innovation which is calculated in the filter for $\mathbf{q}_k^l$. Consequently, to calculate the mixture in (3.34), i.e., to calculate the required posteriors and update their weights, we should run M^k filters.

Given that the number of required filters grows exponentially with time, even in the simple-looking linear and Gaussian situation, calculating the total posterior density is not practical. From the implementation point of view, we need to devise a method where the number of required filters is kept finite. If we consider the mode sequences as branches of a tree, there are two ways to keep the number of branches finite, we should either prune the most unlikely branches or merge similar branches together. Both strategies are approximations which lead to suboptimal solutions. Some examples of the available methods are Multiple Model Pruning (MMP) filter [30,31], Gaussian Pseudo-Bayesian (GPB) filter [31] and Interacting Multiple Model (IMM) filter [30,31]. The first method is based on pruning whereas the other two are merging-based techniques.

3.5.2 First-order Gaussian Pseudo-Bayesian Filter

The idea behind this type of filters is to merge the branches which have the same mode towards the end of the branch but differ in the 'older' modes. The end of the branch denotes the current time. The depth of the filter is defined as the number of consecutive time steps for which the mode sequences should be identical before we start merging the nodes. As the name suggests, the depth of the first order GPB (GPB1) is one, i.e., it considers the modes at the current time and runs M filters each matched to one of the modes of the system. Figure 3.1 depicts one cycle of this filter for a system with two modes.

In this filter, the past information is summarized in a single Gaussian density described by $\hat{\mathbf{x}}_{k-1|k-1}$ and $\mathbf{P}_{k-1|k-1}$. This density is propagated to M filters at time k. The output of each filter is denoted by $\hat{\mathbf{x}}^j_{k|k}$ and $\mathbf{P}^j_{k|k}$. Consequently, the posterior of the state is given by

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}_{1:k}) &= \sum_{j=1}^{M} p(\mathbf{x}_k|\mathbf{z}_{1:k}, q_k = j) Pr\{q_k = j|\mathbf{z}_{1:k}\} \\ &= \sum_{j=1}^{M} p(\mathbf{x}_k|\mathbf{z}_{1:k}, q_k = j)\omega_k^j \end{aligned} \tag{3.36}$$

where ω_k^j is the weight of each mode-conditioned posterior density which is updated by

$$\begin{aligned} \omega_k^j &= Pr\{q_k = j|\mathbf{z}_{1:k}\} \\ &\propto p(\mathbf{z}_k|q_k = j, \mathbf{z}_{1:k-1}) Pr\{q_k = j|\mathbf{z}_{1:k-1}\} \\ &\propto p(\mathbf{z}_k|q_k = j, \mathbf{z}_{1:k-1}) \sum_{i=1}^{M} Pr\{q_k = j|q_{k-1} = i, \mathbf{z}_{1:k-1}\} Pr\{q_{k-1} = i|\mathbf{z}_{1:k-1}\} \\ &\propto p(\mathbf{z}_k|q_k = j, \mathbf{z}_{1:k-1}) \sum_{i=1}^{M} \pi_{ij}\omega_{k-1}^i. \end{aligned} \tag{3.37}$$

At the end of each filtering cycle the Gaussian mixture is approximated by a single Gaussian distribution whose first two moments match that of the mixture. These moments together with the weights of all the modes are saved and sent to the next time step.

3.5.3 Second-order Gaussian Pseudo-Bayesian Filter

The second-order GPB (GPB2) filter considers the mode histories of the current and the previous time step. In this filter the past is summarized

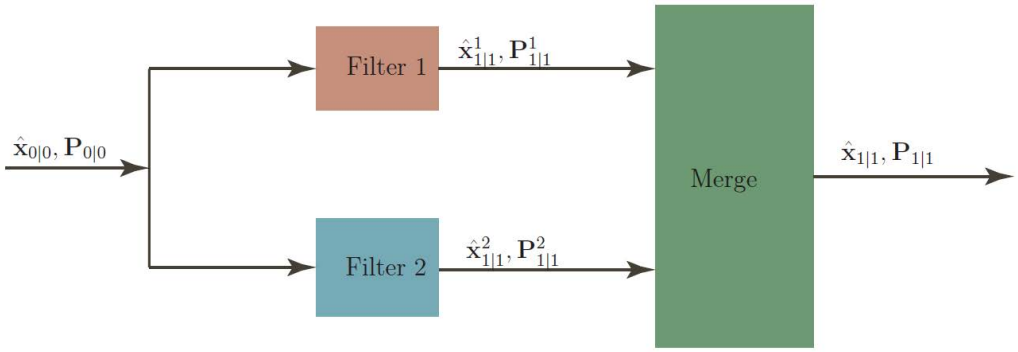

Figure 3.1: Block diagram of one cycle of the GPB1 filter for M=2.

by M mode-conditioned state estimates and their associate covariances, i.e., $\hat{\mathbf{x}}^i_{k-1|k-1}$ and $\mathbf{P}^i_{k-1|k-1}$ where $i = 1, ..., M$. At the beginning of each filter cycle, each mode-conditioned state estimate and its covariance are sent through M mode-matched filters. Posterior densities which are the results of this operation are denoted by

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}, q_k = j, q_{k-1} = i) \approx \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}^{ij}_{k|k}, \mathbf{P}^{ij}_{k|k}). \quad i, j = 1, ..., M$$

Consequently, this algorithm requires M^2 filtering operations. One cycle of this filter for a system with two modes is illustrated in Figure 3.2.

Using the total probability theorem, the posterior density of the state is described by

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}_{1:k}) &= \sum_{j=1}^{M}\sum_{i=1}^{M} p(\mathbf{x}_k|\mathbf{z}_{1:k}, q_k = j, q_{k-1} = i) \\ &\times \; Pr\{q_{k-1} = i|q_k = j, \mathbf{z}_{1:k}\} Pr\{q_k = j|\mathbf{z}_{1:k}\} \\ &= \sum_{j=1}^{M}\sum_{i=1}^{M} p(\mathbf{x}_k|\mathbf{z}_{1:k}, q_k = j, q_{k-1} = i)\omega^{i|j}_{k-1|k}\omega^j_k \end{aligned} \tag{3.38}$$

where ω^j_k is the mode probability which is updated according to

$$\begin{aligned} \omega^j_k &= Pr\{q_k = j|\mathbf{z}_{1:k}\} \\ &\propto p(\mathbf{z}_k|q_k = j, \mathbf{z}_{1:k-1}) \\ &\propto \sum_{i=1}^{M} p(\mathbf{z}_k|q_k = j, q_{k-1} = i, \mathbf{z}_{1:k-1}) \\ &\times \; Pr\{q_k = j|q_{k-1} = i, \mathbf{z}_{1:k-1}\} Pr\{q_{k-1} = i|\mathbf{z}_{1:k-1}\} \\ &\propto \sum_{i=1}^{M} p(\mathbf{z}_k|q_k = j, q_{k-1} = i, \mathbf{z}_{1:k-1})\pi_{ij}\omega^i_{k-1}. \end{aligned} \tag{3.39}$$

Additionally, $\omega_{k-1|k}^{i|j}$ is the merging probability which is calculated by

$$\begin{aligned}\omega_{k-1|k}^{i|j} &= Pr\{q_{k-1}=i|q_k=j,\mathbf{z}_{1:k}\}\\ &\propto p(\mathbf{z}_k|q_k=j,q_{k-1}=i,\mathbf{z}_{1:k-1})\\ &\times Pr\{q_k=j|q_{k-1}=i,\mathbf{z}_{1:k-1}\}Pr\{q_{k-1}=i|\mathbf{z}_{1:k-1}\}\\ &\propto p(\mathbf{z}_k|q_k=j,q_{k-1}=i,\mathbf{z}_{1:k-1})\pi_{ij}\omega_{k-1}^{i}. \end{aligned} \tag{3.40}$$

After the filtering, the state estimate of mode j and its covariance are calculated by the following merging process

$$\begin{aligned}\hat{\mathbf{x}}_{k|k}^{j} &= \sum_{i=1}^{M}\hat{\mathbf{x}}_{k|k}^{ij}\omega_{k-1|k}^{i|j}\\ \mathbf{P}_{k|k}^{j} &= \sum_{i=1}^{M}\omega_{k-1|k}^{i|j}[\mathbf{P}_{k|k}^{ij}\\ &+ (\hat{\mathbf{x}}_{k|k}^{ij}-\hat{\mathbf{x}}_{k|k}^{j})(\hat{\mathbf{x}}_{k|k}^{ij}-\hat{\mathbf{x}}_{k|k}^{j})^T], \end{aligned} \tag{3.41}$$

the results of which are saved at the end of each filtering cycle and sent to the next time step. The output of the filter at time k is calculated by

$$\begin{aligned}\hat{\mathbf{x}}_{k|k} &= \sum_{j=1}^{M}\hat{\mathbf{x}}_{k|k}^{j}\omega_{k}^{j}\\ \mathbf{P}_{k|k} &= \sum_{j=1}^{M}\omega_{k}^{j}[\mathbf{P}_{k|k}^{j}\\ &+ (\hat{\mathbf{x}}_{k|k}^{j}-\hat{\mathbf{x}}_{k|k})(\hat{\mathbf{x}}_{k|k}^{j}-\hat{\mathbf{x}}_{k|k})^T]. \end{aligned} \tag{3.42}$$

Note that these values are not passed on to the next time step but only used as the output of the filter.

3.5.4 Interacting Multiple-Model Filter

At the beginning of an IMM filter cycle, instead of expanding M hypotheses of time $k-1$ to M^2 hypotheses at time k, we use the properties of the mode switching process and perform mode-specific merging. More specifically, we calculate M merged densities according to

$$\begin{aligned}p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1},q_k=j) &= \sum_{i=1}^{M}p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1},q_{k-1}=i,q_k=j)\\ &\times Pr\{q_{k-1}=i|q_k=j,\mathbf{z}_{1:k-1}\}\\ &= \sum_{i=1}^{M}p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1},q_{k-1}=i,q_k=j)\omega_{k-1|k-1}^{i|j} \end{aligned} \tag{3.43}$$

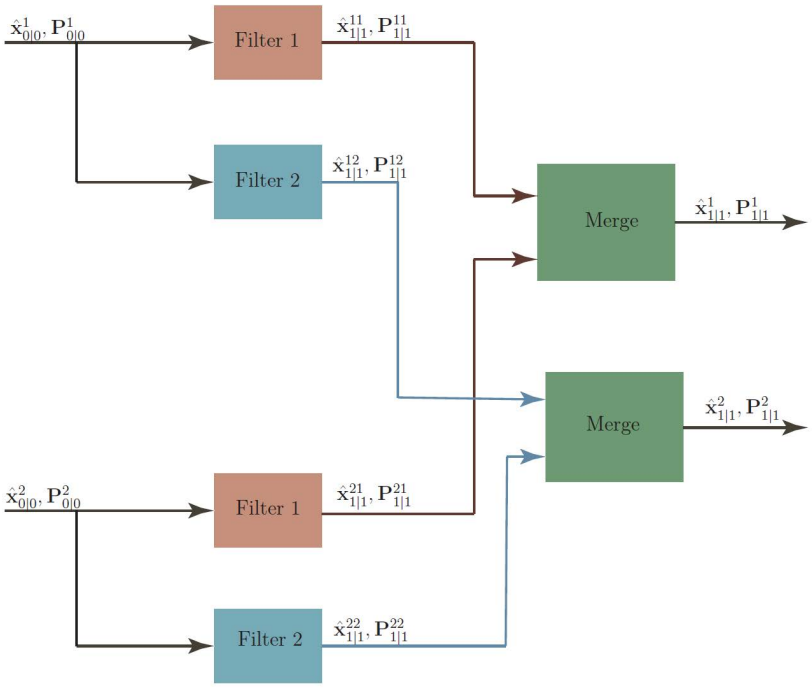

Figure 3.2: Block diagram of one cycle of the GPB2 filter for M=2.

where $\omega^{i|j}_{k-1|k-1}$ is the merging probability given by

$$\begin{aligned} \omega^{i|j}_{k-1|k-1} &= Pr\{q_{k-1} = i | q_k = j, \mathbf{z}_{1:k-1}\} \\ &\propto Pr\{q_k = j | q_{k-1} = i, \mathbf{z}_{1:k-1}\} Pr\{q_{k-1} = i | \mathbf{z}_{1:k-1}\} \\ &\propto \pi_{ij} \omega^i_{k-1}. \end{aligned} \tag{3.44}$$

The fist two moments of each merged density denoted by $\hat{\mathbf{x}}^{(j)}_{k-1|k-1}$ and $\mathbf{P}^{(j)}_{k-1|k-1}$, respectively, are described by

$$\begin{aligned} \hat{\mathbf{x}}^{(j)}_{k-1|k-1} &= \sum_{i=1}^{M} \hat{\mathbf{x}}^i_{k-1|k-1} \omega^{i|j}_{k-1|k-1} \\ \mathbf{P}^{(j)}_{k-1|k-1} &= \sum_{i=1}^{M} \omega^{i|j}_{k-1|k-1} [\mathbf{P}^i_{k-1|k-1} \\ &+ (\hat{\mathbf{x}}^i_{k-1|k-1} - \hat{\mathbf{x}}^{(j)}_{k-1|k-1})(\hat{\mathbf{x}}^i_{k-1|k-1} - \hat{\mathbf{x}}^{(j)}_{k-1|k-1})^T]. \end{aligned} \tag{3.45}$$

This merging step is the key difference between the IMM filter and the GPB2. As a result of this step, the IMM algorithm requires M filter runs and therefore, it merges the mode sequences at the same depth as GPB2 with less computational complexity.

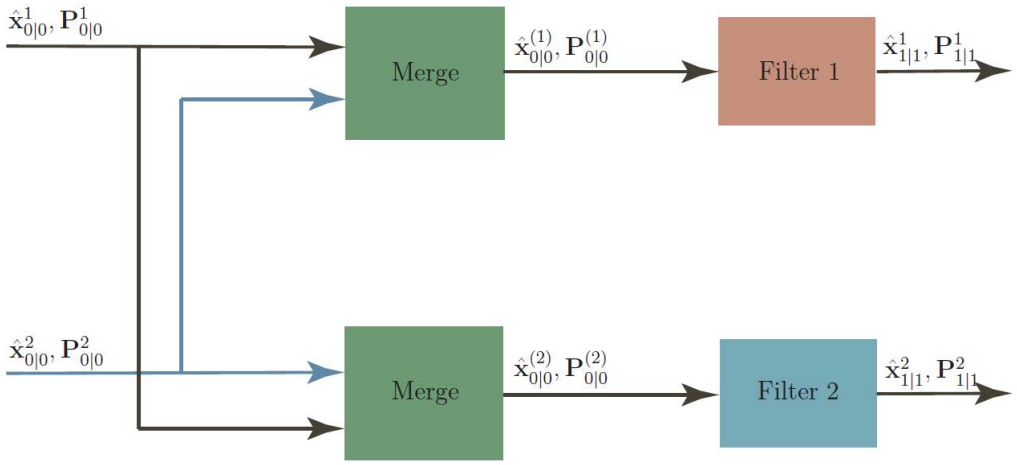

Figure 3.3: Block diagram of one cycle of the IMM filter for M=2.

At the end of the filter cycle, $\hat{\mathbf{x}}^{(j)}_{k-1|k-1}$ and $\mathbf{P}^{(j)}_{k-1|k-1}$ are updated by the filter for mode j which results in the mode-specific output $\hat{\mathbf{x}}^{j}_{k-1|k-1}$ and $\mathbf{P}^{j}_{k-1|k-1}$. Additionally, the mode probabilities are updated by

$$\begin{aligned}
\omega^j_k &= Pr\{q_k = j|\mathbf{z}_{1:k}\} \\
&\propto p(\mathbf{z}_k|q_k = j, \mathbf{z}_{1:k-1}) Pr\{q_k = j|\mathbf{z}_{1:k-1}\} \\
&\propto p(\mathbf{z}_k|q_k = j, \mathbf{z}_{1:k-1}) \sum_{i=1}^{M} p(q_k = j|q_{k-1} = i, \mathbf{z}1 : k-1) p(q_{k-1} = i|\mathbf{z}_{1:k-1}) \\
&\propto p(\mathbf{z}_k|q_k = j, \mathbf{z}_{1:k-1}) \sum_{i=1}^{M} \pi_{ij} \omega^i_{k-1}. \qquad (3.46)
\end{aligned}$$

The posterior density of the state is described by

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \sum_{j=1}^{M} p(\mathbf{x}_k|\mathbf{z}_{1:k}, q^j_k) Pr\{q^j_k|\mathbf{z}_{1:k}\}. \qquad (3.47)$$

which is similar to (3.36) for GPB1. One cycle of the IMM algorithm for a system with $M = 2$ is depicted in Figure 3.3.

It should be noted that the memory storage requirements of IMM is the same as GPB2, i.e., M state estimates, their covariances and their corresponding weights are saved and passed on to the next time step.

Chapter 4

Mapping and Tracking of Multiple Extended Objects

4.1 Problem formulation

Mapping and tracking are closely related concepts. They both refer to the processing of measurements collected by a sensor (or sensors) in order to estimate the states of objects from which the measurements have been generated. We denote the set of measurements, collected at time k, by $\mathbf{Z}_k = \{\mathbf{z}_k^1, \mathbf{z}_k^2, ..., \mathbf{z}_k^{n_k}\}$, where n_k is the number of received measurements at time k. In addition, we denote the set of measurements, collected from time 1 up to and including time k, by $\mathbf{Z}_{1:k} = \{\mathbf{Z}_1, \mathbf{Z}_2, ..., \mathbf{Z}_k\}$. Each measurement can be a vector or a scalar. For example, a radar measures the range and range rate between a sensor and the objects in its field of view. Therefore, a single radar measurement is a vector containing the distance, relative speed and angle to an object.

The objects of interest can be described by a point-object model or an extended-object model. We denote the state of an object at time k by ξ_k and the set of objects at time k by $\mathbf{X}_k = \{\xi_k^1, \xi_k^2, ..., \xi_k^{m_k}\}$, where m_k is the number of objects at time k. The state of interest can contain different quantities depending on the problem at hand.

In both mapping and tracking we seek to estimate $\mathbf{X}_k$ based on $\mathbf{Z}_{1:k}$. In this section, we define mapping and tracking and discuss their similarities and differences. In addition, we describe the object models and explain the conditions that necessitate the use of each model.

4.1.1 Mapping vs. Tracking

The purpose of mapping is to describe an environment as perceived by a sensor. The pose of the sensor is often assumed to be known. The objects

of interest are often referred to as landmarks and are described by their location and some properties,for example, the shape, size and/or color of a landmark. The resulting map is a probabilistic description of the environment, i.e., the map contains quantities of interest as well as uncertainty in those quantities, which is the result of having seen the environment through the lens of a noisy sensor. As such, the state of a landmark contains components that describe its location and components that quantify some property of the landmark.

Mapping problems can be categorized from different aspects. Different categories arise due to, for example, the assumptions regarding the map: static vs. dynamic maps, or the method by which the map is built: sequential vs. batch mapping. In dynamic mapping, the appearance of new landmarks and disappearance of existing landmarks is taken into account via birth/death models [46, 47]. Static maps do not consider appearance/disappearance of the landmarks. In sequential mapping we process the measurements sequentially to estimate the current states of stationary objects in the field of view based on the measurements [46, 47]. In batch mapping, we process all the received measurements, collected at multiple discrete time steps, at the same time to build a map of an area of interest. Paper VI [6] and [14] are examples of batch methods of building static maps.

In tracking, the sensor (sensors) maintains the current state of moving object(s) in its field of view. As such, tracking is often solved as a sequential problem. In this context, the objects of interest are called targets and their states can consist of position, velocity, acceleration, turn-rate, shape and size.

Since targets can appear in the field of view and disappear from it at any time, a complete time-evolution model of the targets takes into account their movements, as well as, birth/death, see for example [48,49]. In Papers V [5] and VII [7], target birth has been taken into account.

4.1.2 Object models

In some applications the distance between the sensor and the object is such that the object covers only one resolution cell of the sensor; i.e., the object looks like a point form the sensor's point of view . Such an object generates at most one measurement per time step, and can be modelled accordingly. Therefore, a point-object model assumes that each object generates at most one measurement per time step. An example of an application where a point-object model is suitable is tracking air planes using ground radar stations.

An extended-object model assumes that an object can generate more

than one measurement per time step. Such a model is suitable in applications where the distance between the sensor and the object are such that the object covers more than one resolution cell of the sensor. For example, an extended-object model is more suitable for tracking/mapping with automotive radars.

In Papers V [5], VI [6], and VII [7], we have modeled the objects as extended objects. Since the focus of this thesis is automotive applications, we will concentrate on mapping/tracking of multiple extended objects in the rest of this Section.

4.1.3 Data association

Data association can be formulated as assigning each of the measurements to an object or clutter and is an important and integral part of any mapping/tracking algorithm. Handling data association in multiple-object tracking/mapping is complicated because of the following intricacies:

I. The probability of detecting an object is often less than one. That is, even if an object is inside the field of view of the sensor, it is not known if it has generated any measurements.

II. It is not known whether the measurements have been generated by an object or they are clutter.

III. The number of objects present in the field of view, and consequently in the area of interest, is unknown.

These difficulties create multiple data association hypotheses, the number of which grows exponentially, or in some cases super exponentially, as the sensor collects more measurements over time. The number of hypotheses is smaller for point-objects since every object can generate at most one measurement per time step. This number is much larger for extended objects, which makes the problem even more complicated.

Addressing the difficulties of mapping/tracking multiple-objects requires models and mathematical tools that take into account uncertainties in the data associations, uncertainties in the number of present objects and uncertainties in the state of each of the objects. Random finite set modeling and finite set statistics [50] provide a rigorous way of addressing these modeling challenges. In Papers V [5], VI [6] and VII [7], we have developed RFS-based algorithms.

4.2 Random finite sets

A random finite set (RFS) is a set whose elements are random variables and whose cardinality is a non-negative integer-valued random variable. Cardinality of set $\mathbf{X} = \{\xi_1, \xi_2, ..., \xi_n\}$ refers to the number of elements in the set, and is denoted by $|\mathbf{X}|$.

Similar to vectors, we can define probability density functions for RFSs. The density of the RFS $\mathbf{X}$ can be expressed by [1]

$$f(\mathbf{X}) = p(n)n! f_n(\xi_1, ..., \xi_n) \tag{4.1}$$

where $p(n)$ is a cardinality distribution, $f_n(\xi_1, ..., \xi_n)$ is a cardinality-conditioned permutation invariant joint distribution. The cardinality distribution is a probability mass function which specifies the probability that the set contains exactly n elements, i.e., $p(n) = \Pr\{|\mathbf{X}| = n\}$.

4.2.1 Overview of RFS-based calculus

In order to incorporate RFSs into Bayesian inference, we need to know how to perform set integration and set differentiation. A set integral is defined as

$$\int \tau(\mathbf{X})\delta\mathbf{X} \triangleq \tau(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int \tau(\{\xi_1, \xi_2, ..., \xi_n\}) d\xi_1 ... d\xi_n, \tag{4.2}$$

and it is used, among other things, to define Probability generating functionals (p.g.fl). P.g.fls provide an alternative representation of an RFS density, one that is often simpler to deal with. The p.g.fl of an RFS density is a transform of the density, defined through the set integral as [50, p. 371]

$$G[h] = \int h^{\mathbf{X}} f(\mathbf{X})\delta\mathbf{X}. \tag{4.3}$$

where $h(\xi)$ is a test function, and

$$h^{\mathbf{X}} \triangleq \begin{cases} \prod_{\xi \in \mathbf{X}} h(\xi) & \mathbf{X} \neq \emptyset \\ 1 & \mathbf{X} = \emptyset \end{cases}.$$

P.g.fls can be regarded as generalized multi-object z-transforms [50]. Basic properties of p.g.fls are [50, p.371]

I. G[h] has no units of measurement;

[1] Throughout this thesis both $p(\cdot)$ and $f(\cdot)$ have been used to denote probability density. From the context it is clear which one is used.

II. $G[0] = f(\emptyset)$;

III. $G[1] = \int f(\mathbf{X})\delta\mathbf{X} = 1$;

IV. $G[h_1] \leq G[h_2]$ if $h_1 \leq h_2$

As an example, consider the density of a union of two independent RFSs $\mathbf{X} = \mathbf{X}_1 \bigcup \mathbf{X}_2$,

$$f(\mathbf{X}) = \sum_{\mathbf{X}_1 \subseteq \mathbf{X}} f_{\mathbf{X}_1}(\mathbf{X}_1) f_{\mathbf{X}_2}(\mathbf{X} - \mathbf{X}_1). \tag{4.4}$$

While this density contains a summation over all possible subsets of the set $\mathbf{X}$, a complex summation to calculate, the corresponding p.g.fl is simply a product [50, p. 372]

$$G_{\mathbf{X}}[h] = G_{\mathbf{X}_1}[h] G_{\mathbf{X}_2}[h]. \tag{4.5}$$

An RFS density can be derived from its p.g.fl by set differentiation [50, pp. 375-376, 384]

$$\begin{aligned} f(\mathbf{X}) &= \frac{\delta}{\delta \mathbf{X}} G[h]|_{h=0} \\ &= \frac{\delta^{|\mathbf{X}|}}{\prod_{\xi \in \mathbf{X}} \delta\xi} G[h]|_{h=0}, \end{aligned} \tag{4.6}$$

where

$$\frac{\delta}{\delta\xi} G[h] \triangleq \lim_{\epsilon \downarrow 0} \frac{G[h + \epsilon\delta_\xi] - G[h]}{\epsilon},$$

and δ_ξ is a Dirac delta function. Generally, p.g.fls are easier to handle than multi-object probability densities; therefore, often in complicated derivations, it is desirable to work with p.g.fls rather than densities. Given a p.g.fl of a set, we can calculate its density using (4.6). This property is reminiscent of the relation between a signal and its Fourier transform (or z-transform).

In signals and systems analysis, we often prefer to perform calculations in the transform domain and perform the inverse transform using well-known pairs. We can follow a similar procedure for RFS densities, i.e., we can perform derivations using p.g.fls (a transform of the density) and find the resulting density using well-known pairs. For example, in the RFS context, the well-known pairs are p.g.fl and density of Poisson, Bernoulli, multi-Bernoulli, etc. processes. For more in depth discussions regarding p.g.fls please consult [50].

Cluster RFS

An RFS whose elements are independent and identically distributed (i.i.d), and which has an arbitrary cardinality distribution, is referred to as a cluster RFS. The multi-object density of a cluster process is given by

$$f(\mathbf{X}) = n!p(n)\prod_{\xi\in\mathbf{X}} f(\xi). \tag{4.7}$$

In addition, the p.g.fl of this process is described as [50]

$$G[h] = G(\langle f; h\rangle), \tag{4.8}$$

where $G(y) = \sum_{n=0}^{\infty} p(n)y^n$ is the probability generating function of the cardinality distribution $p(n)$, and $\langle f; h\rangle = \int f(\xi)h(\xi)d\xi$.

Poisson RFS

A Poisson random finite set is a special kind of cluster RFS where the cardinality is distributed according to a Poisson distribution. The density of a Poisson RFS is given by

$$f(\mathbf{X}) = \exp(-\lambda)\prod_{\xi\in\mathbf{X}} \lambda f(\xi) \tag{4.9}$$

where λ is the expected number of the elements of $\mathbf{X}$, $f(\xi)$ is the density of each of the elements of the set, and $D(\xi) = \lambda f(\xi)$ is referred to as the intensity function. The intensity function is the first moment of the RFS density; a Poisson RFS is completely characterized by its intensity function. The p.g.fl of a Poisson RFS is given by

$$G[h] = \exp(\lambda\langle f; h\rangle - \lambda). \tag{4.10}$$

Bernoulli RFS

A Bernoulli RFS has the following RFS density and p.g.fl,

$$f(\mathbf{X}) = \begin{cases} 1-r, & \mathbf{X} = \emptyset \\ rf(\xi), & \mathbf{X} = \xi \\ 0, & \text{otherwise} \end{cases} \tag{4.11}$$

$$G[h] = 1 - r + r\langle f; h\rangle \tag{4.12}$$

where r is the probability of existence and $f(\xi)$ is the existence-conditioned distribution. The cardinality distribution of a Bernoulli RFS is a Bernoulli distribution with parameter r [49].

A multi-Bernoulli (MB) process is formed by a union of N independent Bernoulli processes, $\mathbf{X} = \bigcup_{i=1}^{N} \mathbf{X}_i$. The p.g.fl of this process, resulting from (4.5) and (4.12), is expressed as

$$G[h] = \prod_{i=1}^{N} (1 - r_i + r_i \langle f_i; h \rangle). \tag{4.13}$$

The RFS density of this process can be described as [16]

$$f(\{\xi_1, \xi_2, ..., \xi_n\}) = \sum_{\uplus_{i=1}^{N} \mathbf{X}_i = \mathbf{X}} \prod_{i=1}^{N} f_i(\mathbf{X}_i). \tag{4.14}$$

where each $f_i(\mathbf{X}_i)$ is a Bernouli distribution of the form described in (4.11).

A multi-Bernoulli mixture (MBM) is an RFS whose density is a normalized weighted sum of densities of multi-Bernoulli (MB) RFSs, that is,

$$f(\mathbf{X}) = \sum_{j} W^j f^j(\mathbf{X}), \tag{4.15}$$

where each weight W^j may be related to, e.g., one data association hypothesis, $\sum_j W^j = 1$, and each $f^j(\mathbf{X})$ has the form presented in (4.14). In addition, the p.g.fl of the MBM process in (4.15) is formed by a weighted sum of the p.g.fl of MB RFSs described by (4.13),

$$G[h] = \sum_{j} W^j \prod_{i} (1 - r^{j,i} + r^{j,i} \langle f^{j,i}; h \rangle). \tag{4.16}$$

4.3 Multi-object Bayes filter

The multi-object Bayes filter is a generalization of the single-sensor single-object Bayes filter. The Kalman filter [34] and the nonlinear filters discussed in Chapter 3 are examples of single-object Bayes filters. The multi-object Bayes filter propagates the target set probability density $f_{k|k-1}(\mathbf{X}_k|\mathbf{Z}_{1:k-1})$ in time using the Bayes update

$$f_{k|k}(\mathbf{X}_k|\mathbf{Z}_{1:k}) = \frac{f_k(\mathbf{Z}_k|\mathbf{X}_k) f_{k|k-1}(\mathbf{X}_k|\mathbf{Z}_{1:k-1})}{\int f_k(\mathbf{Z}_k|\mathbf{X}_k) f_{k|k-1}(\mathbf{X}_k|\mathbf{Z}_{1:k-1}) \delta\mathbf{X}_k} \tag{4.17}$$

and the Chapman-Kolmogorov prediction

$$f_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k}) = \int f_{k+1,k}(\mathbf{X}_{k+1}|\mathbf{X}_k) f_{k|k}(\mathbf{X}_k|\mathbf{Z}_{1:k}) \delta\mathbf{X}_k, \tag{4.18}$$

where $f_k(\mathbf{Z}_k|\mathbf{X}_k)$ is the multi-object measurement set density which contains the underlying models of detection and false alarms. In addition, $f_{k+1,k}(\mathbf{X}_{k+1}|\mathbf{X}_k)$ is the multi-object transition density that describes the dynamic behavior of the objects. This density includes motion models describing movements of objects, birth models accounting for the appearance of new objects into the field of view and survival/death models describing the disappearance of objects from the field of view. Measurement and dynamic models are discussed in more detail in Section 4.4.

4.3.1 Approximate multi-object filters

The set integrals in (4.17) and (4.18) are computationally intractable. In the remainder of this section, we discuss approximate multi-object Bayes filters.

Probability hypothesis density Filter

Probability hypothesis density (PHD) refers to the first-order multi-object moment, also called intensity function. The PHD filter [51] is an approximation of the multi-object Bayes filter. Instead of the multi-object posterior density $f_{k|k}(\mathbf{X})$, the PHD filter propagates the posterior intensity $D_{k|k}(\xi)$ in time. In each recursion, the PHD filter predicts and updates the intensity function, and approximates the multi-object density as a Poisson process after both prediction and update, based on the following assumptions:

A.I Objects evolve and generate measurements independent of one another.

A.II The RFS of surviving objects is independent of the birth RFS.

A.III Clutter measurements are independent of object-generated measurements.

A.IV Clutter RFS is Poisson.

These assumptions are common in most tracking applications.

The point target PHD filter has desirable computational properties. At any given time step, its computational complexity is $O(m_k n_k)$, where m_k is the number of objects and n_k is the number of measurements in the set $\mathbf{Z}_k$. This filter simultaneously associates all measurements with all tracks, therefore, avoiding the combinatorial problem of enumerating and ranking a list of possible associations [50].

One disadvantage of the PHD filter arises due to assumed Poisson distribution for the number of objects. Since the mean and variance of a

Poisson distribution are equal, this filter has a large uncertainty regarding the number of objects when many objects are present in the scene. This high variance causes the cardinality estimates to be very sensitive to the presence of false alarms and missed detections [50]. The replacement of the full distribution by its first moment causes a large information loss; however, the loss can be overcome if signal-to-noise ratio (SNR) is high enough [50].

Sequential Monte Carlo (SMC) and Gaussian-mixture versions of the PHD filter have been developed in [52–54] for point objects. Additionally, the PHD filter for extended objects is presented in [55].

Cardinalized PHD filter

The cardinalized PHD filter (CPHD) alleviates the information loss of the PHD filter by propagating the full cardinality distribution $p_{k|k}(n)$ of $f_{k|k}(\mathbf{X})$, as well as the posterior PHD $D_{k|k}(\xi)$, in time [50]. Therefore, each recursion of the CPHD includes predicting and updating both the PHD and the cardinality distribution.

The CPHD filter relies on assumptions A.I to A.III and relaxes the rest of the PHD assumptions to:

B.I The clutter RFS is an i.i.d cluster process and independent of the RFS of object-generated measurements.

B.II The prior and predicted multi-object RFSs are i.i.d cluster processes.

B.III The cardinality of the multi-object set has an arbitrary distribution.

Assumption B.III is the most important difference between the PHD and CPHD filters. This difference is part of the reason why the CPHD filter maintains a more accurate and stable estimate of the number of objects [50]. The disadvantage of the point target CPHD filter is that it has a computational complexity of $o(m_k n_k^3)$ which is higher than the PHD filter.

The CPHD filter was developed in [56]; its analytic implementation was presented in [57], a version of this filter which incorporates spawning was proposed in [58], and a CPHD filter for extended objects was presented in [59].

Multi-Bernoulli filters

The orginal multi-object multi-Bernoulli (MeMBer) filter was proposed in [50], along with a Gaussian-mixture implementation of it. Unlike PHD and CPHD filters, where the recursions include propagating the multi-object first moment, the MeMBer filter propagates an approximate multi-object density. The introduced approximations rely on having a low false alarm

rate, and are shown to result in a large cardinality bias. The cardinality balanced MeMBer (CMeMBer) filter which corrects this bias is presented in [49].

A closed from solution, i.e., a conjugate prior form, to the multi-object Bayes recursion in (4.17) and (4.18) is presented in [60] and implemented in [61]. This filter incorporates labeled multi-Bernoulli RFSs. In [16] a hybrid Poisson multi-Bernoulli mixture (PMBM) filter for point objects is discussed. This filter proposes yet another closed form solution to the multi-object Bayes recursion without requiring labels for the underlying RFSs. Note that in the context of multi-object tracking conjugacy does not imply tractability and this is discussed in detail in [5]. Each recursion of the PMBM filter includes prediction and update of the parameters of the Poisson MBM. The Poisson part describes the distribution of the objects that have not been detected so far, and the MBM part describes the detected objects. Each of the summands of the MBM corresponds to a data association hypothesis. This summation has exponential computational complexity as a function of the number of measurements collected over time, and [16] proposes approximations that yield tractable tracking algorithms. Further, in [16], it is shown that the proposed methods outperform CPHD and CMeMBer, especially in cases with low probability of detection, for a similar computational cost.

In Paper V [5] we have shown that the PMBM is a conjugate prior for the Poisson measurement model of extended objects (see Section 4.4.1 for the description of this measurement model), and in Paper VI [6] we have used this form to describe the batch multi-object posterior density of the map. Additionally, in Paper VII [7] we develop a Poisson multi-Bernoulli filter that recursively approximates the PMBM of the posterior with a Poisson multi-Bernoulli distribution.

4.4 Modeling

An integral part of the multi-object Bayes filter in (4.17) and (4.18) is the measurement model and the transition (time-evolution) model. In a mapping/tracking algorithm, these underlying models are chosen so that they match the type of object of interest. In this chapter we discuss models that are suitable for tracking/mapping extended objects.

4.4.1 Measurement model

A commonly used measurement model for multiple extended objects is the inhomogeneous Poisson model developed in [62], which is based on single

object Poisson modelling presented in [63]. Papers V [5], VI [6] and VII [7] describe the extended object measurements by an inhomogeneous Poisson process.

In this section we give an overview of the Poisson measurement model for single and multiple extended objects and some of the models that describe the structure of an extended object.

Single-object Poisson model

A measurement can be generated by multiple sources on a single extended object. Modelling the sources explicitly requires evaluating probabilities of all measurement to source associations. Such detailed description is challenging especially when the number of sources and their characteristics are highly uncertain. The Poisson measurement model does not require explicit modeling of measurement sources. In this model, an object is represented by a spatial probability distribution, and each measurement is an independent random draw from the spatial model. At each time step, the number of measurements originating from an object is modeled by a Poisson distribution. Furthermore, clutter measurements are also modeled by a Poisson process, and they are assumed to be generated independently from the object-originated measurements. Accordingly, the measurement model of the vector $\mathbf{z}_k = [\mathbf{z}_k^1, ..., \mathbf{z}_k^{n_{z,k}}]$ is given by

$$\begin{aligned} p(\mathbf{z}_k|\xi_k) &= p(n_{z,k}|\xi_k)p(\mathbf{z}_k|\xi_k, n_{z,k}) \\ &= \frac{e^{-\lambda_k}{\lambda_k}^{n_{z,k}}}{n_{z,k}!}\prod_i p(\mathbf{z}_k^i|\xi_k), \end{aligned} \tag{4.19}$$

where λ_k is the measurement model Poisson rate and can be assumed to be constant, or can be included in the state vector.

Multi-object inhomogeneous Poisson model

In this model, the measurements over the sensor field of view are represented by an inhomogeneous Poisson process. The probability of n measurements being in the region $\mathcal{A}$ of the field of view is

$$\Pr\{n_{z,k}(\mathcal{A}) = n\} = \frac{e^{-\lambda_k(\mathcal{A})}\lambda_k(\mathcal{A})^n}{n!}, \tag{4.20}$$

where

$$\lambda_k(\mathcal{A}) = \int_{\mathcal{A}} D_k(\mathbf{z}_k^i|\xi_k)dz \tag{4.21}$$

is the expected number of measurements in region $\mathcal{A}$, and $D_k(\mathbf{z}_k^i|\xi_k)$ is the spatially dependent intensity function of the Poisson process. The likelihood of each measurement in region $\mathcal{A}$ is proportional to the intensity function in this region,

$$p(\mathbf{z}_k^i|\xi_k) = \frac{D_k(\mathbf{z}_k^i|\xi_k)}{\lambda_k(\mathcal{A})}. \tag{4.22}$$

That is, given the state ξ_k, each measurement is a random draw from the probability distribution in (4.22).

4.4.2 Models for shape and size of an extended object

The spatial model of an extended object should describe the structure of that object. The shape and size of an extended object are represented by the state of an object ξ_k. For example, [64] proposes a state ξ_k that is a combination of a kinematic vector $\mathbf{x}_k \in \mathbb{R}^{n_x}$ and a positive semi-definite matrix X_k describing the extent of the object, i.e., $\xi_k = (\mathbf{x}_k, X_k)$. A consequence of this model is that the shape of an object is assumed to be elliptical. Furthermore, in [64], the kinematic state is assumed to have a Gaussian distribution and the extent is assumed to have an inverse Wishart distribution. This combination is a conjugate prior for the measurement model given by

$$p(\mathbf{z}_k|\xi_k) = \mathcal{N}(\mathbf{z}_k; H\mathbf{x}_k, X_k), \tag{4.23}$$

where H is a known linear measurement model. Using (4.23) with a Gaussian inverse Wishart prior results in a linear measurement update. In the measurement model described in (4.23), the measurements are assumed to be spread accross the object surface, and the measurement covariance is described by the extent of the object. Consequently, the sensor measurement noise is assumed to be negligible compared to the extent of an object.

In [65,66], the extent of an object as well as the sensor measurement noise are accounted for; however, the resulting updates will be non-linear and have to be approximated. Besides the models discussed so far, the kinematic states and the parameters governing the shape and size of an object can be incorporated into one state vector ξ_k. This allows for modelling the extent by shapes other than an ellipsoid. For a detailed discussion regarding different methods of modelling the structure of an extended object please see [67].

In Papers V [5], VI [6] and VII [7], we have described the extent of the objects by random matrices and assumed a Gaussian inverse Wishart prior for them. Moreover, we have included the Poisson rate in the state vector

and described it by a gamma prior which is a conjugate prior for the Poisson rate.

4.4.3 Time-evolution model

The prediction step of the multi-object Bayes filter is the solution to (4.18). This step relies on an RFS model of the time evolution of the object set $\mathbf{X}_k$ which describes the appearance of new objects (birth), disappearance of objects from the scene (survival/death), and motion of each surviving object. Accordingly, the set $\mathbf{X}_k$ is expressed as a union of two independent sets,

$$\mathbf{X}_k = \mathbf{X}_k^s \bigcup \mathbf{X}_k^b, \tag{4.24}$$

where $\mathbf{X}_k^s$ is the RFS of surviving objects and $\mathbf{X}_k^b$ is the RFS of newly born objects. This independence implies that objects are born independent of surviving objects. The model of $\mathbf{X}_k^b$ is problem dependent. Two common ways of describing birth processes are Poisson and multi-Bernoulli models. The Poisson birth model has been used in, e.g., [52, 54, 56, 68] for point objects and in, e.g., [58, 69, 70] for extended objects. In [49] object birth is modeled as a multi-Bernoulli process. Existing objects survive form time $k-1$ to k with state dependent probability of survival $p_s(\xi_{k-1})$. More specifically, the dynamic behaviour of an object at time k can be described as an RFS $\mathbf{X}_k^s(\xi_{k-1})$ that is equal to $\{\xi_k\}$ with probability $p_s(\xi_{k-1})$, or is empty with probability $1-p_s(\xi_{k-1})$. Accordingly, $\mathbf{X}_k^s = \bigcup_{\xi_{k-1}\in\mathbf{X}_{k-1}} \mathbf{X}_k^s(\xi_{k-1})$. Each surviving object transitions from ξ_{k-1} to ξ_k according to the transition model $p(\xi_k|\xi_{k-1})$.

The transition model for an extended object describes the motion model of both kinematic states and the extent states. The transition for the kinematic states is often described by

$$\xi_k = f(\xi_{k-1}) + w_k, \tag{4.25}$$

where $f(\cdot)$ can be a linear or nonlinear motion model and w_k is a Gaussian noise. For a survey on motion models for manoeuvring objects see [71]. The transition model for the extent state can be modelled in several different ways. In [64, 66, 72], it is assumed that the transition of the extent state, where the extent is modelled by a random matrix, is independent from the kinematic state. While this assumption holds for constant velocity and constant acceleration motion models, it does not hold for a constant/variable turn-rate model, a.k.a, the coordinated turn model. Because, in a turning maneuver, the turning of the extent is a function of the turn-rate which is

a part of the kinematic state. A transition model that allows the extent transition to be dependent on kinematic states is presented in [73].

In Papers V [5] and VII [7], the extent transition is described by the model presented in [73] and object birth is modeled by a Poisson process.

Chapter 5

Road Geometry Estimation

5.1 Introduction

This chapter focuses on road geometry estimation and some of the underlying road models. In addition, we will discuss some of the functions of advanced driver assistant systems (ADAS) which require road geometry information.

Road geometry estimation refers to estimating the shape (course) of the road ahead of the host vehicle up to a certain distance using the available information provided by on-board sensors. Three examples of the road course are the shape of the middle of the host vehicle's lane, the shape of the parallel lane markings denoting the borders of the host vehicle's lane and the shape of the road barriers, e.g., guardrails, which define the borders of the road. Observations provided by different on-board sensors can include measurements of lane markings provided by a camera, measurements of other vehicles and stationary objects by the road provided by a radar.

This estimation problem can be formulated as a Bayesian filtering problem. In this context, the state vector at time k denoted by $\mathbf{r}_k$, contains all the information required to describe the road at that time and is referred to as the road state. The elements of this vector depend on the road model and the chosen parametrization. The complete solution to the filtering problem at time k is the posterior density of the road state given all the observations up to and including time k, i.e., $p(\mathbf{r}_k|\mathbf{z}_{1:k})$. Furthermore, the time evolution of the road state is defined by a process model and the sensor observations are described by a measurement model. If all of these models are linear and the uncertainties are Gaussian, then the solution can be found by the Kalman filter. However, if either/all of the models are nonlinear, a suboptimal solution can be found by applying a nonlinear filtering algorithm, e.g., EKF, UKF, CKF, etc.

5.2 Motivation and Applications

Most advanced driver assistance systems (ADAS) rely on a perception of the host vehicles's environment. The road geometry is one element of this perception which, among other things, enables us to answer three important questions: 1) where is the host vehicle's lane? 2) is the leading vehicle/obstacle in the host vehicle's lane? 3) where are the road barriers?

The goal of the safety functions of ADAS is to support the driver in dangerous situations to prevent an accident or mitigate its consequences. The support provided by a safety function can vary between warning of an impending danger to partly/completely taking control of the vehicle. Some safety functions focus on supporting the driver in dangerous situations that could occur at close range, e.g., lane departure warning (LDW) [74]. As a result, the distance up to which such applications require the road geometry is relatively low (50-60m). On the other hand applications which aim at earlier interventions at higher speeds, e.g., collision avoidance by steering (CAbS) [75] and forward collision warning/avoidance (FCW/A) [76], need to know the road geometry at longer distances.

In this section we will give a brief overview of some of the ADAS functions and explain why each of them requires road geometry information.

Lane Guidance Systems

The purpose of these systems is to prevent lane departures. Their level of support varies between generating an audible warning to applying a torque to the steering wheel (intervention). They include a lane-position-sensing algorithm which normally uses the camera as the sensing modality.

To determine if the host vehicle is about to cross the lane markings, the system needs to predict the trajectory of the vehicle a few time steps ahead. For example in a highway scenario, for a time to lane crossing (TLC) of 1 second, the shape of the lane markings up to 30-40m ahead is required [77].

Adaptive Cruise Control

Adaptive cruise control (ACC) [78] is considered a comfort function. Such systems automatically adjust the host vehicle's speed to maintain a safe distance to the leading vehicle which is in the same lane as the host vehicle. ACC systems are mainly intended for use in highways, however, the stop and go version of this application is designed to support the driver in congested traffic.

If an ACC system is to determine which of the surrounding vehicles is the leading vehicle in the host lane, it requires the shape and width of the host lane ahead of the host vehicle. For example if the ACC radar detects a vehicle at 150m ahead of the host vehicle and their relative speed is such that the system needs to adapt the host speed, then the road geometry up

to that distance is required in order to decide if that vehicle is on the host lane.

Curvature Warning
One dangerous situation which could cause an accident is entering a sharp curve with too high a speed. Curvature warning applications [75] address this situation by warning the driver if the host vehicle is approaching a sharp curve too fast. To detect such an event, the application requires the knowledge of the shape of the road ahead of the host vehicle. A sufficiently early warning assists the driver to reduce the speed by a comfortable rate. The distance up to which the road geometry information is needed depends on the velocity reduction required for safe entrance to the curve as well as the deceleration rate of the braking.

The calculation of the required distance can best be clarified by an example. We assume that the maximum deceleration rate by which passengers of a car will feel comfortable is $2m/s^2$ [79]. If a driver using the curvature warning system is to reduce the velocity of the vehicle by 20Kph, the system should generate a warning 3.8 seconds before the curve entrance. This includes the driver reaction time which can be assumed to be 1s. If the initial velocity of the vehicle is 100kph, then the system requires the shape of the road at a greater distance than 98m ahead.

Forward Collision Warning/Avoidance
Forward collision warning/avoidance systems (FCW/A) aim at supporting the driver to avoid a rear-end accident or to mitigate its consequences. They do so by generating a warning if the host vehicle is approaching a slower leading vehicle with too high speed. Additionally, if the driver does not react adequately, the system brakes automatically. These systems are mainly designed for highways where due to high velocities of the vehicles, a timely intervention/warning is crucial. Examples of FCW/A systems are proposed in [80–84].

In order to generate effective warnings, i.e., to keep the false alarm rate low while maintaining a high detection rate, the system needs an accurate knowledge of the road geometry among other things. This knowledge is required for correct lane assignment. An FCW/A algorithm should distinguish the leading vehicle in the host vehicle's lane early enough, so that a correct and timely warning can be generated. According to the results provided by [80] and [82], the road geometry at distances longer than 100m is required.

5.3 Road Estimation Systems

Driver assistance systems and autonomous driving, which are two main applications of road geometry estimation, have been an active topic of research for more than two decades and consequently, a large body of work precedes this thesis. This section attempts to provide a broad picture of the road estimation systems proposed in the literature over this period.

Road geometry estimation can be carried out using different sensor set ups. The job of the available sensors is to provide the estimation algorithm with information from the road environment. Within this context, two categories of road geometry estimation methods can be observed: short-range road geometry estimation, i.e., up to 50m ahead of the host vehicle and long-range, i.e., 100m or longer ahead. While the former problem is rather mature and has been studied for many years [77], the latter has not been studied as extensively. Accordingly, the road geometry estimation papers fit into three groups: 1) vision-based systems, 2) radar/Lidar-based systems and 3) fusion systems where the measurements of different sensors are fused together. Among these three, the papers in the first group aim at estimating the road at short-range, while the papers in the second and third group estimate the road up to longer distances.

5.3.1 Vision-based systems

Since the road environment is adapted to the visual system of a human driver, using a camera to extract the same cues seems like a natural solution. Besides this, the fact that most early vision-based systems seek to extract the shape of the lane markings, a feature that is best sensed by a camera, could explain why early works on road geometry estimation are vision-based. Additionally, high information content, low cost and low power consumption are among the reasons why the vision-based methods are so extensively investigated.

Vision-based methods estimate the road geometry by detecting visual cues such as lane markings and road edges in images provided by either a monocular or stereo camera. Such images present a 2D or 3D description of the scene which will be used to estimate a 2D/3D road model. A comprehensive survey of vision-based methods can be found in [77]. Examples of systems where the road is modeled in 2D can be found in [85–91]. Additionally, [92–94] use a 3D road model within their road geometry estimation algorithm.

Occlusions, poor lighting, limited visibility due to adverse weather conditions or ambient light, etc, can pose serious difficulties for a purely vision-based system. Furthermore, the poor effective resolution of the lane mark-

ings at far distances in the image provided by the camera [95], makes a camera-only system an unsuitable choice for applications such as ACC [78], where the road geometry up to a longer distance is required.

5.3.2 Radar/Lidar-based systems

While advantages of using the camera modality provide appealing arguments for vision-based methods, the limitations of it drove the researchers to look for other sensors such as the radar or laser-based radar (lidar/ladar). A radar can sense objects up to a longer distance (200m), cope with adverse weather conditions and is not susceptible to ambient light. Examples of radar-only systems are [96–98]. These systems estimate the road using measurements of the road boundary, e.g., stationary objects by the road, provided by the radar. Lidar has higher resolution than radar but can not be used in extreme weather conditions. Examples of Lidar-based systems can be found in [99] and [100].

5.3.3 Fusion Systems

The main drawback of the radar is that it can not detect the lane markings. Therefore, while the camera can provide us with lane marking information in near distances under good weather conditions, the radar lacks this capability. On the other hand, the radar can provide us with measurements of the position and velocity of other vehicles on the road and the position of stationary objects by the road, up to longer distances. From this perspective, the characteristics of the camera and the radar seem to compliment each other. Hence, fusing the information provided by the camera and the radar has the potential to improve the road estimation performance.

Measurements of other vehicles provided by a radar have been fused with the measurements of lane markings received from a camera in [101–105]. These papers make use of the assumption that the leading vehicles follow their lane and do not change lane or take an exit. Under this assumption, the movement of the leading vehicles gives valuable information about the course of the road ahead. Furthermore, Paper II [2] of this thesis is another example where measurements of the lane markings are fused with those of the leading vehicles. However in this paper, instead of assuming that the leading vehicles follow their lane, we try to detect the motion state of these vehicles and distinguish the ones which follow their lane from those that do not.

Apart from other vehicles, the position of stationary objects by the road can be a valuable source of information. An example of fusing the lane marking measurements and measurements of road boundaries can be found

in [106]. Additionally, in [95] and Paper III [3] of this thesis, we present a road geometry estimation algorithm where measurements of the lane markings, leading vehicles and stationary objects are fused together to improve the road geometry estimation.

Information about the shape of the road included in the digital maps can also be used to estimate the road geometry. In [107], the authors present a method where the map data is combined with the information provided by a vision-based lane detection system to estimate the road ahead.

5.4 Road Model

A road model describes the shape of the road up to a certain distance from the host vehicle. Roads are usually built according to the guidelines which are to ensure a smooth ride for the driver. More specifically, they are built such that they can be driven with constant and smooth steering wheel movements [108]. Among other things, this indicates that the curvature of the road should not change abruptly [109]. The geometrical shape of roads is often a combination of straight lines, circular curves and transition curves [110]. The transition curves are normally designed using clothoids which are parametric curves where the curvature changes linearly with the arc length, i.e.,

$$\kappa_r(s) = \kappa_0 + \kappa_1 s \tag{5.1}$$

where $\kappa_r(s)$ is the curvature of the road at arc length s, κ_0 is the curvature at the origin and κ_1 is the curvature change rate. Note that $\kappa_r(s) = \frac{1}{R(s)}$, where $R(s)$ is the radius of the turn of the road at arc length s.

The clothoid model is often used for describing the road in the horizontal plane. The 2D geometrical shape of the road is summarized by a single clothoid. This model can be described in a Cartesian coordinate system by

$$x_r(s) = x_0 + \int_0^s \cos(\varphi_r(s))ds \tag{5.2}$$

$$y_r(s) = y_0 + \int_0^s \sin(\varphi_r(s))ds \tag{5.3}$$

where

$$\varphi_r(s) = \varphi_0 + \kappa_0 s + \frac{\kappa_1}{2}s^2 \tag{5.4}$$

is the heading at arc length s. The position, heading and curvature of the road at the initial point, are denoted by x_0, y_0, φ_0 and κ_0, respectively.

Additionally, $x_r(s)$ and $y_r(s)$ are road coordinates at arc length s. Examples of Cartesian coordinate systems are local Cartesian coordinate system, i.e., a coordinate system attached to the host vehicle and global Cartesian coordinate system.

5.4.1 Third Degree Polynomial Road Model

The integrals in (5.2) and (5.3) do not have closed form solutions. Therefore, in order to calculate them we need to resort to approximations. A commonly used approximation in the literature which describes the road in a local Cartesian coordinate system is

$$x_r(s) = x_0 + \int_0^s \cos(\varphi_r(s))ds \approx s \tag{5.5}$$

$$y_r(s) = y_0 + \int_0^s \sin(\varphi_r(s))ds \approx y_0 + \varphi_0 s + \frac{\kappa_0}{2}s^2 + \frac{\kappa_1}{6}s^3 \tag{5.6}$$

where x_0, y_0, φ_0 and κ_0 are the position, heading and curvature of the road at the position of the host vehicle. Furthermore, $x_0 = 0$, since it refers to the x-coordinate of the origin of the local Cartesian coordinate system.

Examples of methods where the third degree polynomial road model is used can be found in [92, 93, 96, 105, 111, 112]. This model is widely used for the situations where the sensing modality is a camera as well as for the applications where the host vehicle requires the shape of the road up to 50-60 meters ahead.

The approximation in (17) and (5.6) assumes that $\varphi_r(s)$ is sufficiently small so that $\sin(\varphi_r(s)) \approx \varphi_r(s)$ and $\cos(\varphi_r(s)) \approx 1$. Consequently, arc length is approximated by x. The result of this approximation is a third degree polynomial model. According to [93] and [92], the approximation in (17) is valid for small heading changes, i.e., $|\Delta\varphi_r(s)| = |\varphi_r(s) - \varphi_0| \leq \alpha$, where $\alpha \simeq 11^\circ - 15^\circ$. In other words, the approximations that are used to derive this model are valid up to $s = s_{max}$ where s_{max} is the arc length up to which the heading change is small. Therefore, although the polynomial model is a powerful model for describing the road at close range, the length up to which it is capable of describing the road accurately, is limited.

The limitation of the third degree polynomial model can be illustrated using an example where the road in the field of view of the sensors includes segments with two different curvature change rates. We assume that the road within the field of view includes a straight line ($\kappa_0 = 0$ and $\kappa_1 = 0$) followed by a transition curve, i.e., a clothoid. Furthermore, we assume that the transition curve is designed for a 90kph road. The recommended

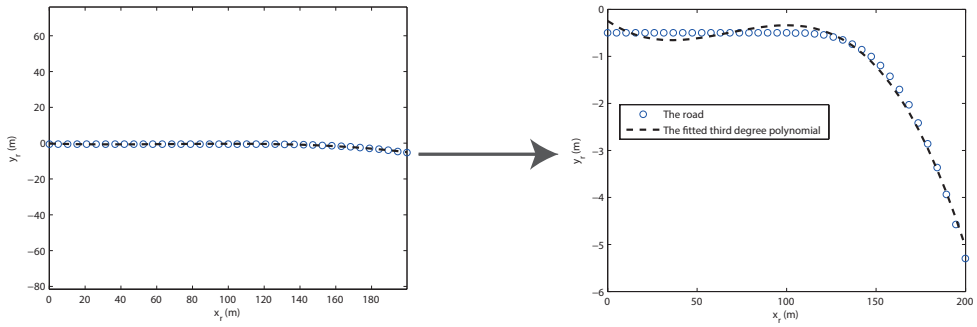

Figure 5.1: Comparison between the road and the third degree polynomial fitted to it. The right-hand side figure uses vertical zoom to clarify the comparison.

maximum curvature change rate for this road is determined by

$$\kappa_1 = \frac{\dddot{V}}{V^3} \tag{5.7}$$

where V is the velocity and $\dddot{V} = 0.45m/s^3$ is the maximum jerk [113]. The length of the constructed road is 200 meters and the length of each segment is 100 meters. To see how the polynomial model can describe this road, we fit a third degree polynomial to the road by using least squares. The constructed road and the fitted third degree curve are depicted in Figure 5.1. A close look at this figure reveals that the close range accuracy is sacrificed so that the fitted curve can capture the curvature of the second segment. This clearly shows that the third degree polynomial model is not suitable for modeling long stretches of roads.

Additionally, for describing stretches of roads which include more than one segment, a single clothoid is not an adequate model to begin with. To describe the road in these situations a segmented model where each segment can capture the characteristics of different segments of actual roads is a suitable alternative. Such a model should also accommodate the fact that real roads are smooth structures.

5.4.2 Spline-based Road Models

Splines are piecewise polynomial curves which are widely used to approximate smooth geometrical shapes, e.g., curves and surfaces. One commonly used spline type is Basis-splines (B-splines). In this family of spline functions, the spline is generated by a linear combination of a set of basis func-

tions, i.e.,

$$\mathbf{f}(t) = \sum_{i=0}^{N-1} \mathbf{p}_i B_i^p(t) \quad t_{min} \leq t \leq t_{max} \tag{5.8}$$

where $\mathbf{f}(t)$ is the curve defined by the set of control points $\mathbf{p}_i$ which can be derived by imposing interpolation/approximation conditions on the given data points [114]. Additionally, $B_i^p(t)$ is the ith basis function of degree p and t is the independent variable along which the curve is defined.

The basis functions are defined on a vector of real numbers called knots, denoted by $\mathbf{u} = [u_0, u_1, ..., u_{n_K}]$ where $u_j \leq u_{j+1}$, $u_j \in [t_{min}, t_{max}]$ and $n_K = N + p$. The jth B-spline basis function of degree p is recursively calculated by [114]

$$B_j^0(t) = \begin{cases} 1 & u_j \leq t < u_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

$$B_j^p(t) = \frac{t - u_j}{u_{j+p} - u_j} B_j^{p-1}(t) + \frac{u_{j+p+1} - t}{u_{j+p+1} - u_{j+1}} B_{j+1}^{p-1}(t) \quad t \in [u_j, u_{j+p+1}).$$

From the definition we can see that each basis function is active (nonzero) in part of the parameter span. As a result, at any given interval of the curve, only a subset of all the basis functions is active. Moreover, the knots can be seen as the points within the curve where the set of active basis changes.

Figure 5.2 illustrates a set of cubic B-spline basis functions defined on $\mathbf{u} = [0, 0, 0, 0, 1, 2, 4, 7, 7, 7, 7]$. Knots can have multiplicity larger than one. The curve generated by B-splines, i.e., $\mathbf{f}(t)$ is differentiable infinite times at the interior of the knot intervals. Furthermore, at a knot with multiplicity of k, $\mathbf{f}(t)$ is $p - k$ times differentiable. As a result, if all the interior knots are distinct ($k = 1$), a third degree (cubic) B-spline provides us with second order continuity. In this example the curve is defined over $t \in [0, 7]$ and the multiplicity of the first and last knot is $p + 1 = 4$.

To illustrate how well a B-spline can model a road we revisit the previous example. This time the road is approximated by a cubic B-spline. Figure 5.3 depicts the comparison between the cubic B-spline and the third degree polynomial fit to the two segmented road. It can be seen that the cubic B-spline captures the shape of the road more accurately.

Road models based on B-splines have been introduced in the literature, e.g., [90] and [115]. The road model presented in Papers II [2] and III [3] can be described as a spline formed by joining clothoids with G^2-continuity, i.e., the position, heading and curvature of two segments are equal at their joints. Additionally, the road model presented in [95] approximates the road as a 2D curve by discrete points.

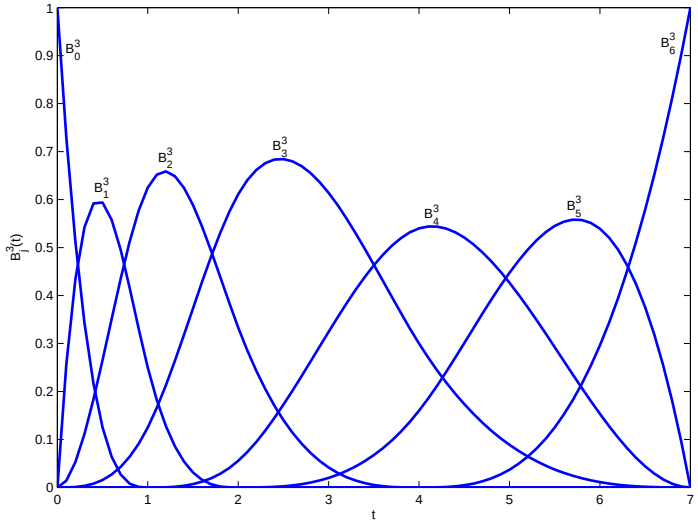

Figure 5.2: Cubic B-spline basis functions.

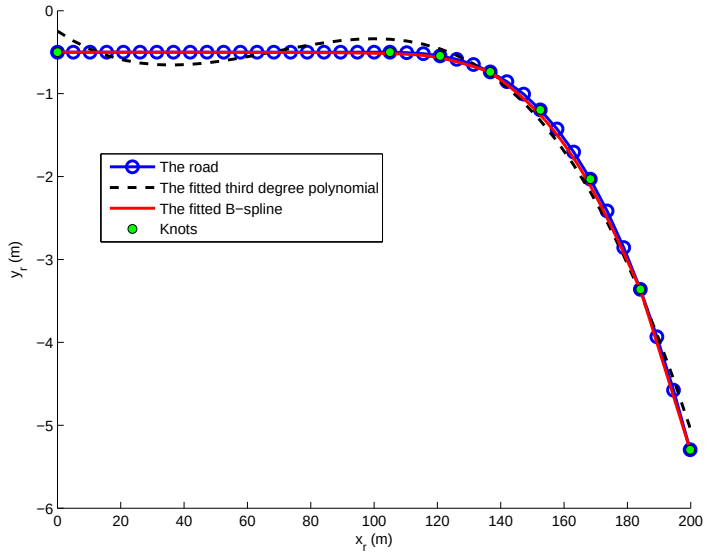

Figure 5.3: Comparison of B-spline and third degree polynomial fit to a two-segmented road

Bibliography

[1] M. Fatemi, L. Svensson, L. Hammarstrand, and M. Morelande, "A study of map estimation techniques for nonlinear filtering," in *2012 15th International Conference on Information Fusion (FUSION)*, pp. 1058–1065.

[2] M. Fatemi, L. Hammarstrand, L. Svensson, and Á. F. García-Fernández, "Road geometry estimation using a precise clothoid road model and observations of moving vehicles," in *2014 17th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 238–244.

[3] L. Hammarstrand, M. Fatemi, Á. F. García-Fernández, and L. Svensson, "Long-range road geometry estimation using moving vehicles and roadside observations," *Accepted for publication in IEEE Trans. Intell. Transp. Syst.*

[4] M. Fatemi, L. Svensson, L. Hammarstrand, and M. Lundgren, "Variational Bayesian EM for SLAM," in *2015 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2015, pp. 501–504.

[5] K. Granström, M. Fatemi, and L. Svensson, "Poisson multi-bernoulli conjugate prior for estimation of both detected and undetected extended objects," *Submitted to IEEE Trans. Signal Process.*

[6] M. Fatemi, K. Granström, L. Svensson, F. J. Ruiz, and L. Hammarstrand, "Poisson multi-Bernoulli radar mapping using Gibbs sampling," *Submitted to IEEE Trans. Signal Process.*

[7] M. Fatemi, L. Svensson, and K. Granström, "Poisson multi-bernoulli filter for extended object tracking," *to be submitted.*

[8] K. Granström, M. Fatemi, and L. Svensson, "Gamma Gaussian inverse-Wishart Poisson multi-Bernoulli Filter for Extended Target

Tracking," in *2016 19th Conference on Information Fusion (FUSION)*.

[9] Á. F. García-Fernández, L. Hammarstrand, M. Fatemi, and L. Svensson, "Bayesian road estimation using onboard sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1676–1689, 2014.

[10] M. Fatemi, H. Amindavar, and J. A. Ritcey, "Noise reduction via harmonic estimation in Gaussian and non-Gaussian environments," *Signal Processing*, vol. 90, no. 5, pp. 1554–1561, 2010.

[11] WHO. (2016, May) Road traffic injuries. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs358/en

[12] C. Mui, "Fasten Your Seatbelts: Google's Driverless Car Is Worth Trillions (Part 1)," *Forbes*, 2013.

[13] N. A. Greenblatt, "Self-Driving Cars Will Be Ready Before Our Laws Are," *IEEE Spectrum*, 2016.

[14] M. Lundgren, L. Svensson, and L. Hammarstrand, "Variational bayesian expectation maximization for radar map estimation," *IEEE Trans. Signal Process.*, vol. 64, no. 6, pp. 1391–1404, 2016.

[15] J. L. Williams, "Hybrid Poisson and multi-Bernoulli filters," in *2012 15th International Conference on Information Fusion (FUSION)*, pp. 1103–1110.

[16] ——, "Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA and association-based MeMBer," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 3, pp. 1664–1687, 2015.

[17] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[18] S. Jain and R. M. Neal, "A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model," *Journal of Computational and Graphical Statistics*, vol. 13, no. 1, pp. 158–182, 2004.

[19] Á. F. García-Fernández, L. Svensson, and M. R. Morelande, "Multiple target tracking based on sets of trajectories," *arXiv preprint arXiv:1605.08163*.

[20] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[21] C. M. Bishop, *Pattern recognition and machine learning*. New York, USA: Springer, 2006.

[22] M. I. Jordan, *Learning in graphical models*. Springer Science & Business Media, 1998, vol. 89.

[23] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 6, pp. 721–741, 1984.

[24] G. Casella and E. I. George, "Explaining the Gibbs sampler," *The American Statistician*, vol. 46, no. 3, pp. 167–174, 1992.

[25] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, University of London, 2003.

[26] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[27] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[28] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Mathematical Statistics*, pp. 79–86, 1951.

[29] Z. Ghahramani and M. J. Beal, "Propagation algorithms for variational Bayesian learning," *Advances in neural information processing systems*, pp. 507–513, 2001.

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

[31] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.

[32] G. Sibley, G. Sukhatme, and L. Matthies, "The iterated sigma point Kalman filter with applications to long range stereo," in *Online Proc. 2nd Robotics: Science and Systems Conf.*, Philadelphia, PA, Aug 2006.

[33] Á. F. García-Fernández, L. Svensson, M. R. Morelande, and S. Särkkä, "Posterior linearization filter: principles and implementation using sigma points," *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5561–5573, 2015.

[34] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, pp. 35–45, 1960.

[35] R. van der Merwe and E. A. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[36] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Trans. Autom. Control*, vol. 54, no. 6, pp. 1254–1269, June 2009.

[37] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[38] A. H. Stroud, *Approximate calculation of multiple integrals*. Prentice-Hall Englewood Cliffs, NJ, 1971, vol. 431.

[39] N. Andreasson, A. Evgrafov, and M. Patriksson, *An Introduction to Continuous Optimization*. Studentlitteratur, 2005.

[40] B. M. Bell and F. W. Cathey, "The iterated Kalman filter update as a gauss-newton method," *IEEE Trans. Autom. Control*, vol. 38, no. 2, pp. 294–297, 1993.

[41] D. G. Luenberger, *Linear and nonlinear programming*. Kluwer Academic Publishers, 2003.

[42] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[43] R. L. Bellaire, E. W. Kamen, and S. M. Zabin, "A new iterated filter with applications to target tracking," in *SPIE Signal and Data Processing of Small Targets*, San Diego, CA, USA, 1995.

[44] Á. F. García-Fernández and L. Svensson, "Gaussian MAP filtering using Kalman optimization," *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1336–1349, 2015.

[45] M. R. Morelande and Á. F. Garcia-Fernandez, García-Fernández, "Analysis of Kalman filter approximations for nonlinear measurements," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5477–5484, 2013.

[46] M. Adams, B.-N. Vo, R. Mahler, and J. Mullane, "SLAM gets a PHD: New concepts in map estimation," *IEEE Robot. Autom. Mag.*, vol. 21, no. 2, pp. 26–37, 2014.

[47] H. Deusch, S. Reuter, and K. Dietmayer, "The labeled multi-Bernoulli SLAM filter," *Signal Processing Letters, IEEE*, vol. 22, no. 10, pp. 1561–1565, 2015.

[48] R. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, 2003.

[49] B.-T. Vo, B.-N. Vo, and A. Cantoni, "The cardinality balanced multi-target multi-Bernoulli filter and its implementations," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 409–423, 2009.

[50] R. P. Mahler, *Statistical multisource-multitarget information fusion*. Artech House, Inc., 2007.

[51] R. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, 2004.

[52] B.-N. Vo, S. Singh, and A. Doucet, "Sequential monte carlo implementation of the PHD filter for multi-target tracking," in *Proc. Int. Conf. on Information Fusion*, 2003, pp. 792–799.

[53] ——, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224–1245, 2005.

[54] B. Vo and W. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 54, no. 11, p. 4091, 2006.

[55] K. Granström, C. Lundquist, and U. Orguner, "A Gaussian mixture PHD filter for extended target tracking," in *2010 13th Conference on Information Fusion (FUSION)*, 2010, pp. 1–8.

[56] R. Mahler, "PHD filters of higher order in target number," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1523–1543, 2007.

[57] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3553–3567, 2007.

[58] M. Lundgren, L. Svensson, and L. Hammarstrand, "A CPHD filter for tracking with spawning models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 496–507, June 2013.

[59] C. Lundquist, K. Granstrom, and U. Orguner, "An extended target CPHD filter and a gamma Gaussian inverse Wishart implementation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 472–483, 2013.

[60] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3460–3475, 2013.

[61] B.-N. Vo, B.-T. Vo, and D. Phung, "Labeled random finite sets and the Bayes multi-target tracking filter," *IEEE Trans. Signal Process.*, vol. 62, no. 24, pp. 6554–6567, 2014.

[62] K. Gilholm, S. Godsill, S. Maskell, and D. Salmond, "Poisson models for extended target and group tracking," in *Optics & Photonics 2005*. International Society for Optics and Photonics, 2005, pp. 59 130R–59 130R.

[63] K. Gilholm and D. Salmond, "Spatial distribution model for tracking extended objects," in *IEEE Proceedings on Radar, Sonar and Navigation*, vol. 152, no. 5, 2005, pp. 364–371.

[64] J. W. Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 3, pp. 1042–1059, 2008.

[65] M. Feldmann and D. Fränken, "Tracking of extended objects and group targets using random matrices: a new approach," in *2008 11th International Conference on Information Fusion*, 2008, pp. 1–8.

[66] M. Feldmann, D. Fränken, and W. Koch, "Tracking of extended objects and group targets using random matrices," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1409–1420, 2011.

[67] K. Granström and M. Baum, "Extended object tracking: Introduction, overview and applications," *arXiv:1604.00970*.

[68] R. Mahler, "The multisensor PHD filter, i: General solution via multitarget calculus," in *Proc. of SPIE*. IEEE, 2009.

[69] K. Granström, C. Lundquist, and O. Orguner, "Extended target tracking using a Gaussian-mixture PHD filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 4, pp. 3268–3286, 2012.

[70] K. Granström and U. Orguner, "A PHD filter for tracking multiple extended targets using random matrices," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5657–5671, 2012.

[71] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. part i. dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, 2003.

[72] J. Lan and X. R. Li, "Tracking of extended object or target group using random matrix–part I: New model and approach," in *2012 15th International Conference on Information Fusion (FUSION)*, pp. 2177–2184.

[73] K. Granström and U. Orguner, "New prediction for extended targets with random matrices," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 2, pp. 1577–1589, 2014.

[74] M. Chen, T. Jochem, and D. Pomerleau, "AURORA: A vision-based roadway departure warning system," in *1995 IEEE/RSJ International Conference on Intelligent Robots and Systems.'Human Robot Interaction and Cooperative Robots'*, vol. 1. IEEE, 1995, pp. 243–248.

[75] A. Eidehall, J. Pohl, F. Gustafsson, and J. Ekmark, "Toward autonomous collision avoidance by steering," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 84–94, 2007.

[76] N. H. T. S. Administration, "Development of a collision avoidance system," *US department of Transportation*, 2005.

[77] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20–37, 2006.

[78] H. Winner, "Adaptive cruise control system: Aspects and development trends," *Overview and update of ITS system developments*, 1996.

[79] Z. Wu, Y. Liu, and G. Pan, "A smart car control model for brake comfort based on car following," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 42–46, 2009.

[80] T. Butsuen, T. Niibe, T. Takagi, Y. Yamamoto, H. Seni *et al.*, "Development of a rear-end collision avoidance system with automatic brake control," *JSAE Rev.*, vol. 15, no. 4, pp. 335–340, 1994.

[81] Y. Fujita, K. Akuzawa, and M. Sato, "Radar brake system," *JSAE Review*, vol. 16, no. 2, 1995.

[82] H. Araki, K. Yamada, Y. Hiroshima, and T. Ito, "Development of rear-end collision avoidance system," in *1996 Intelligent Vehicles Symposium (IV)*, pp. 224–229.

[83] P. Barber and N. Clarke, "Advanced collision warning systems," 1998.

[84] P. Seiler, B. Song, and J. K. Hedrick, "Development of a collision avoidance system," *Development*, vol. 4, pp. 17–22, 1998.

[85] K. Kluge, "Extracting road curvature and orientation from image edge points without perceptual grouping into features," in *1994 Intelligent Vehicles Symposium (IV)*. IEEE, 1994, pp. 109–114.

[86] D. Pomerleau, "RALPH: Rapidly adapting lateral position handler," in *1995 Intelligent Vehicles Symposium (IV)*, pp. 506–511.

[87] K. Kluge and S. Lakshmanan, "A deformable-template approach to lane detection," in *1995 Intelligent Vehicles Symposium (IV)*. IEEE, 1995, pp. 54–59.

[88] M. Bertozzi and A. Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. Image Process.*, vol. 7, no. 1, pp. 62–81, 1998.

[89] R. Aufrere, R. Chapuis, and F. Chausse, "A model-driven approach for real-time road recognition," *Machine Vision and Applications*, vol. 13, no. 2, pp. 95–107, 2001.

[90] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using b-snake," *Image and Vision computing*, vol. 22, no. 4, pp. 269–280, 2004.

[91] Y. Wang, L. Bai, and M. Fairhurst, "Robust road modeling and tracking using condensation," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 4, pp. 570–579, 2008.

[92] E. D. Dickmanns and B. D. Mysliwetz, "Recursive 3-D road and relative ego-state recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 199–213, 1992.

[93] E. D. Dickmanns and A. Zapp, "A curvature-based scheme for improving road vehicle guidance by computer vision," in *Cambridge Symposium_ Intelligent Robotics Systems*. International Society for Optics and Photonics, 1987, pp. 161–168.

[94] R. Chapuis, R. Aufrere, and F. Chausse, "Accurate road following and reconstruction by computer vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 4, pp. 261–270, 2002.

[95] A. F. García-Fernández, L. Hammarstrand, M. Fatemi, and L. Svensson, "Bayesian road estimation using on-board sensors," *IEEE Transaction on intelligent transportation systems*, vol. 15, no. 4, pp. 1676–1689, 2014.

[96] K. Kaliyaperumal, S. Lakshmanan, and K. Kluge, "An algorithm for detecting roads and obstacles in radar images," *IEEE Trans. Veh. Technol.*, vol. 50, no. 1, pp. 170–182, 2001.

[97] Y. Yamaguchi, M. Sengoku, and S. Motooka, "Using a van-mounted fm-cw radar to detect corner-reflector road-boundary markers," *IEEE Trans. Instrum. Meas.*, vol. 45, no. 4, pp. 793–799, 1996.

[98] C. Lundquist, L. Hammarstrand, and F. Gustafsson, "Road intensity based mapping using radar measurements with a probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1397–1408, 2011.

[99] W. S. Wijesoma, K. S. Kodagoda, and A. P. Balasuriya, "Road-boundary detection and tracking using ladar sensing," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 456–464, 2004.

[100] K. Peterson, J. Ziglar, and P. E. Rybski, "Fast feature detection and stochastic parameter estimation of road shape using multiple lidar," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 612–619.

[101] Z. Zomotor and U. Franke, "Sensor fusion for improved vision based lane recognition and object tracking with range-finders," in *1997 Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 1997, pp. 595–600.

[102] A. Gern, U. Franke, and P. Levi, "Advanced lane recognition-fusing vision and radar," in *2000 Intelligent Vehicles Symposium (IV)*. IEEE, 2000, pp. 45–51.

[103] A. Eidehall, J. Pohl, and F. Gustafsson, "Joint road geometry estimation and vehicle tracking," *Control Engineering Practice*, vol. 15, no. 12, pp. 1484–1494, 2007.

[104] U. Hofmann, A. Rieder, and E. D. Dickmanns, "Radar and vision data fusion for hybrid adaptive cruise control on highways," *Machine Vision and Applications*, vol. 14, no. 1, pp. 42–49, 2003.

[105] C. Lundquist and T. B. Schön, "Joint ego-motion and road geometry estimation," *Information Fusion*, vol. 12, no. 4, pp. 253–263, 2011.

[106] B. Ma, S. Lakshmanan, and A. O. Hero, "Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 3, pp. 135–147, 2000.

[107] C. Gackstatter, S. Thomas, and G. Klinker, "Fusion of clothoid segments for a more accurate and updated prediction of the road geometry," in *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Funchal, Sep. 2010, pp. 1691 – 1696.

[108] C. Lundquist, "Sensor fusion for automotive applications," Ph.D. dissertation, Linköping University, 2011.

[109] A. Gern, U. Franke, and P. Levi, "Robust vehicle tracking fusing radar and vision," in *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2001, pp. 323–328.

[110] R. Lamm, B. Psarianos, T. Mailaender *et al.*, *Highway design and traffic safety engineering handbook*. McGraw-Hill New York, 1999, vol. 2.

[111] C. Lundquist and T. Schon, "Road geometry estimation and vehicle tracking using a single track model," in *2008 Intelligent Vehicles Symposium (IV)*, pp. 144–149.

[112] A. Hernandez-Gutierrez, J. I. Nieto, T. Bailey, and E. M. Nebot, "Probabilistic road geometry estimation using a millimetre-wave radar," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 4601–4607.

[113] A. Eidehall and F. Gustafsson, "Combined road prediction and target tracking in collision avoidance," in *2004 Intelligent Vehicles Symposium (IV)*. IEEE, 2004, pp. 619–624.

[114] L. Biagiotti and C. Melchiorri, *Trajectory planning for automatic machines and robots*. Springer, 2008.

[115] H. Loose and U. Franke, "B-spline-based road model for 3-D lane recognition," in *2010 13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2010, pp. 91–98.