Thesis for the degree of Doctor of Philosophy

Bayesian Filtering for Automotive Applications

by

Malin Lundgren

CHALMERS

Department of Signals and Systems
Chalmers University of Technology
Göteborg, Sweden 2015

Bayesian Filtering for
Automotive Applications
Malin Lundgren
ISBN 978-91-7597-175-9

This thesis has been prepared using LaTeX.

Doktorsavhandlingar vid Chalmers Tekniska Högskola
ISSN 0346-718X
Series No 3856

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

Printed by Chalmers Reproservice
Göteborg, Sweden, March 2015

To Ellen and Edith

Abstract

This thesis is concerned with how data from common automotive sensors can be processed and interpreted in order to support advanced driver assistance systems (ADAS). More specifically, the thesis addresses aspects of object tracking using radar detections, mapping and self-localization for automated vehicles and driver monitoring.

In automotive radar tracking, an observed vehicle typically generates multiple detections. This thesis presents a detailed sensor model that adapts to the detection properties of an object by jointly estimating the position of reflection centres and the position of the object. Moreover, the model considers the limited resolution of the radar and evaluation show results close to those achieved with a deterministic vehicle model where the reflecting properties are known. A second contribution to the area of object tracking is a generalization of the well-known cardinalized probability hypothesis density (CPHD) filter to incorporate objects that appear through spawning from existing targets. It is further shown that the generalized filter is tractable for some common birth and spawning models.

For automated vehicles, some of the studied problems resemble those traditionally studied in robotics, such as mapping and localization. This thesis presents and evaluates a self-localization solution based on a set of automotive off-the-shelf sensors together with a map that contains lane markings and a simplistic description of radar landmarks. The evaluation shows that this map, in combination with real radar data, provides valuable information to the localization algorithm. With this motivation, a method for estimating more detailed radar maps is derived. The map is modelled by an inhomogeneous Poisson process describing the expected measurements from the static environment as a function of the sensor position. The estimation principle relies on a variational method where the number of landmarks and their respective parameters are found simultaneously.

In addition to sensors that observe the vehicle and its surroundings, there are camera-based systems designed to monitor the driver behavior. In the context of driver distraction, this thesis presents a method for driver gaze zone estimation, i.e., estimation of which area the driver is currently looking at, using data provided by such monitoring systems. To improve robustness, the proposed solutions make use of functions that describe the gaze direction based on the head pose and eye closure. It is also shown how these functions can be learnt from data.

Acknowledgement

So much have happened during these years and many people have, directly or indirectly, been a part of the process resulting in this thesis. I would like to start by thanking Prof. Mats Viberg for giving me the opportunity to join the Signal Processing group. To current and former members of the SP group, thank you for contributing to an open and relaxed atmosphere despite the stress that we all feel from time to time.

My deepest thanks goes to my supervisor Lennart Svensson. I really appreciate your enthusiastic and supportive approach to supervision! And thank you for all the inspiring meetings and discussions regarding various ideas/issues/results/theories, but also regarding how to conduct research. To my co-supervisor Lars Hammarstrand, thank you for your support and for a great cooperation during many years. Who would have guessed when we started our Master's thesis project, that we would end up in neighbouring offices working together.

During my years as a Ph.D. student I have been very lucky to be surrounded by talented people who are working on similar topics. When I joined the SP group it was a privilege to have Daniel, Lars and Fredrik around for questions and discussions (ranging from filtering to särskrivningar). Thanks to Abu, Maryam and Erik for keeping the tracking/filtering spirit high, and a big thank you for proofreading this thesis.

I would also like to thank Ana and Maria for all long (but often not long enough) lunches. These moments are very valuable for me. Last but definitely not least: To Dad and Åsa, Mum and Lars, thank you for everything. To my sisters and best friends, Anna and Sandra, thanks for always being there. I love you! And to Ellen, Edith and Kristoffer, I'm so grateful to have you in my life and I love you so much!

Malin Lundgren
Göteborg, March, 2015

List of Publications

This thesis is based on the following publications:

Paper I
L. Hammarstrand, M. Lundgren and L. Svensson. "Adaptive Radar Sensor Model for Tracking Structured Extended Objects". In *IEEE Transactions on Aerospace and Electronics*, vol. 48, no. 3, July 2012

Paper II
M. Lundgren, L. Svensson and L.Hammarstrand. "A CPHD Filter for Tracking With Spawning Models". In *IEEE Journal of Selected Topics in Signal Processing, Special Issue on Multitarget Tracking* , vol. 7, no. 3, June 2013

Paper III
M. Lundgren, E. Stenborg, L. Svensson and L. Hammarstrand. "Vehicle Self-localization Using Off-the-shelf Sensors And a Detailed Map". In *IEEE Intelligent Vehicle Symposium Proceedings*, Dearborn, USA, June 2014

Paper IV
M. Lundgren, L. Svensson and L. Hammarstrand. "Variational Bayesian Expectation Maximization for Radar Map Estimation". Submitted to *IEEE Transactions on Signal Processing*

Paper V
M. Lundgren, L. Hammarstrand and T. McKelvey. "Driver Gaze Zone Estimation Using Bayesian Filtering and Gaussian Processes". Submitted to *IEEE Transactions on Intelligent Transportation Systems*

Contents

Part I
Introduction

Chapter 1

Introduction

The areas of automotive safety and comfort systems have expanded a lot over the years, in order to provide the driver with a comfortable and safe driving experience. The safety development started with passive systems such as seat belts and, later on, air bags, with the aim to protect the driver and the passengers in case of an accident. Systems that are designed to support the driver during travel is referred to as advanced driver assistance systems (ADAS). This category includes, for example, active safety systems such as collision warning and autonomous braking, driver distraction/drowsiness warning as well as autonomous driving functions.

In order to operate, many ADAS systems require knowledge about the surrounding environment, the current traffic situation and the driver. This knowledge includes information regarding the position and movement of vehicles and pedestrians, the location of stationary objects and the shape of the road ahead. To acquire this information, the vehicle is equipped with sensors that observe the surroundings, the driver or properties of the host vehicle. Common sensors in automotive settings are: radars, cameras and internal sensors, such as gyroscopes and accelerometers. However, due to imperfections, these sensors provide noisy observations of the measured quantities and to mitigate these effects it is common to filter the sensor output.

Bayesian filtering provides a natural framework for dealing with noisy observations, combining data from multiple sensors and quantifying the uncertainties in the resulting estimates. The idea in Bayesian filtering is to recursively estimate an unknown quantity over time. This quantity is called the state vector and is observed, directly or indirectly, by one or several sensors. Each filter recursion consists of two steps, namely a prediction step, followed by a measurement update. In the prediction step, a future value of the state is forecasted based on a process (or motion) model. This model captures the behaviour of the state over time, including uncertainties. In the measurement update, the predicted state is updated using new information provided by the sensors. To relate the received measurements to the state, the filter requires a measurement (or sensor) model. Since the measurements are noisy, the updated estimate will also be associated with uncertainties.

In many scenarios there are practical issues that must be considered in order to perform filtering. For example, in addition to the noise that affects the accuracy of the observations, a sensor can either fail to detect an object or provide false alarms. Some sensors, such as radars, also have a limited field of view and a limited resolution. Furthermore, in object tracking, there might be multiple objects in the observed scenario, requiring a robust method for assigning the received measurements to the tracked objects. There is also a need to handle the appearance and disappearance of objects.

In this thesis the focus is on Bayesian methods for how data from common automotive sensors can be processed and interpreted in order to support advanced driver assistance systems. More specifically, the thesis addresses aspects of object tracking using radar detections, mapping and self-localization for automated vehicles and driver monitoring. The thesis is divided into two parts. The first part summarizes the main contributions and put them into context. It also includes a theoretical background to used concepts and methods. In the second part, the research contributions of the thesis is presented in the form of five appended papers.

1.1 Research projects

The work presented in this thesis has been carried out within projects financed by the Intelligent Vehicle Safety System (IVSS) Program and the Strategic Vehicle Research and Innovation Program (FFI) which is funded by the Swedish Agency for Innovation Systems (VINNOVA).

1.2 Contributions of the thesis and future directions

The appended papers cover a range of problems related to how data from common automotive sensors can be processed in order to extract information for ADAS systems. Next, the main contributions of the appended papers are summarized.

Paper I: Adaptive Radar Sensor Model for Tracking Structured Extended Objects

Traditionally, in object tracking using radar measurements the objective has been to track aircrafts at a far distance, mainly for military applications. In this setting, a detected object is much smaller than the resolution of the sensor, implying that the object can be treated as a point target that at most generates one detection at each time step. When tracking vehicles at short distances, the situation is different. Then, the sensor typically resolves multiple reflecting features on the same object, and the number of features varies depending on the distance and angle from which the object is observed. In tracking theory, objects that generate multiple detections at each time are called extended objects.

In this paper we derive a detailed sensor model for tracking of extended objects. The model is based on a line structure that holds a set of reflection centers whose number and positions are unknown. In the tracking framework, the model is adapted to describe the received measurements. Evaluation is performed on real radar data and the achieved tracking performance is compared to that of a detailed model where the size of the car and its reflection properties are known. The derived solution provides similar results as the reference model but at a higher computational cost.

Paper II: A CPHD Filter for Tracking With Spawning Models

In many applications where the aim is to perform object tracking, there are multiple objects to consider and often the number of objects in the observed region changes over time. A framework that naturally deals with multiple objects, including appearance and disappearance, is based on finite set statistics (FISST). In contrast to traditional tracking solutions, the objects are represented by sets where the number of elements is a discrete random variable.

The two most well-known filters, derived using FISST, are the probability hypothesis density (PHD) filter and the cardinalized probability hypothesis density filter (CPHD). In these filters, the process model includes a birth process and a death/survival process that model the appearance and disappearance of objects. In the PHD filter, new objects can be modelled in two ways. Either, they appear spontaneously, i.e. independently of other objects, or they can spawn from existing objects. In the original derivations of the CPHD filter, no spawning process is included.

In this paper we generalize the CPHD filter to include a model of spawning objects. Incorporation of the spawning process only affects the process model, and consequently, the measurement update equations remain the same as for the original CPHD filter. It is further shown that the prediction of the intensity function can be adopted from the PHD filter while a new expression for prediction of the cardinality distribution is required. The derived prediction equation can be intractable due to its combinatorial form. However, we show that for common birth and spawning models, exact or approximate expressions can be derived to make the filter tractable.

Paper III: Vehicle Self-Localization Using Off-the-Shelf Sensors And a Detailed Map

In the work towards self-driving vehicles, it is of a great importance to have detailed information regarding the position of the ego vehicle. This problem is called self-localization and is often solved using a combination of a detailed map and sensors that observe properties of the vehicle as well as the surrounding environment. In many demonstrations that have taken place, expensive and bulky sensors have been used.

In this work we use a set of production sensors to investigate their potential for self-localization. The solution is based on a forward looking radar, a camera system, a GPS, a gyroscope and wheel speed sensors. As common in automotive settings, the camera system delivers extracted features, in this case polynomials that describe the lane markings. Similar, the radar provides filtered detections describing the range and angle to stationary objects. To be able to relate the received measurement from the extrospective sensors to features of known position in the world, a simplistic map of the road as seen by the radar and the camera is generated. This map enables the mapping of camera and radar measurements to features of known positions.

When the vehicle is travelling on road segments where there are both lane markings and good radar landmarks, the evaluation shows a longitudinal position accuracy within 1 meter and a lateral accuracy around 0.2 meters. However, when this is not the case, the algorithm is not robust enough to provide stable and accurate position estimates.

Paper IV: Variational Bayesian Expectation Maximization for Radar Map Estimation

There are many examples of maps in the literature, and many of them represents landmarks as points in the observed space. In this paper we propose that a radar map for localization should describe the behaviour of the detections rather than the position of point sources (which is not what the environment looks like). The proposed model of the map is similar to that in Paper III but where each landmark is described by its own parameters. That is, each landmark is represented by a position in the east-north coordinate frame, a covariance matrix that capture the extension and a weight that corresponds to the expected number of detections from the landmark. In addition, the map includes a description of the clutter intensity. Together, the environment is described as an inhomogeneous Poisson process.

To estimate the map, we derive two batch solutions, using the expectation maximization (EM) algorithm and variational Bayesian EM (VBEM). Both solutions estimate the landmark parameters and the clutter intensity while considering the unknown data associations. However, a major advantage with the VBEM compared to EM is its ability to estimate the number of landmarks jointly with their parameters. To limit the influence of the measurement noise on the landmark extensions estimates, the variation in the noise due to the distance and the angle between the sensor and a landmark is incorporated in the models. Unfortunately, incorporation of the measurement noise results in a model that is not straightforward to use in the VBEM framework. To overcome this issue, a set of approximations are employed. Altogether, the proposed mapping algorithm is straightforward to implement, computationally efficient and shows promising results.

Paper V: Driver Gaze Zone Estimation Using Bayesian Filtering And Gaussian Processes

It is well-known that many accidents are caused by driver inattention. It can be a driver who is using a phone, looking at things besides the road or talking to a passenger. With the aim to detect if the driver is tired or distracted, there are camera-based monitoring systems designed to observe the driver. These systems provide measurements on, for example, the head pose, the gaze direction and the eye opening. In this paper, we use such information to find the probability that the driver is looking at different gaze zones. In particular, the focus is on a set of zones directly related to active driving and to distraction, for example: the road, the mirrors and down at a display or a hand-held phone.

In the paper, we consider three sets of information commonly provided by driver monitoring systems and a solution is derived for each set. For systems that do not provide observations of the gaze, it is shown how a probabilistic function that describe the gaze direction based on the head pose and eye closure can be incorporated into a Bayesian filter. The function can also be beneficial as a complement to noisy and unstable gaze data. It is further shown how such a function can be estimated from a set of data with known gaze focus using Gaussian processes.

1.2.1 The author's contributions

Paper II, IV and V are mainly the author's own work. Of course, theory, different ideas, encountered issues and structure of the papers have been discussed with the co-authors. Paper I is mainly Lars Hammarstrand's work. The author of this thesis took part in various aspects of the development, implementation and writing. Finally, Paper III is the result of a close cooperation with Erik Stenborg. The author of this thesis contributed to all parts (development, implementation and writing).

1.2.2 Future work

The discussion regarding future work is focused on the topics of the two most recent papers in the thesis, namely radar mapping and driver monitoring/distraction.

Radar mapping

When observed by a radar, many objects have different properties depending on the distance and the angle from which they are detected. In the extreme case, an object might be fully observable from some directions while occluded from other directions. Several objects might also be unresolved when observed from some distances but resolved from others. Incorporation of this type of information would provide an even better description of the detections from the environment.

When estimating radar maps using simulated data, there are many possible measures that can be considered in order to evaluate the proposed methods. However, when using real data, no ground truth is available for evaluation. Hence, to be able to evaluate such a map, there is a need for an alternative method. Possible approaches are to evaluate the map regarding the ability to describe new data or the ability to support another function, such as a localization algorithm.

Driver monitoring/distraction

In Paper V in this thesis, it is shown how the incorporation of a gaze mapping in the filtering framework can improve the ability to determine what the driver is looking at. To further improve the description of the gaze direction, it is possible to include additional information in the mapping. Example of such inputs can be head rotational speed or signals from the vehicle, such as steering, buttons etc.

Access to information from additional sensors or systems also makes it possible to extend the proposed gaze zone estimation algorithm to consider dynamic zones. When having zones describing the road as well as vehicles and pedestrians, the algorithm can be used to answer questions such as: has the driver seen the pedestrian approaching the road ahead? is the driver aware of the car in the blind spot region? etc. Worth noticing is that since the exact positions of other objects are not known, the corresponding gaze zones will be associated with uncertainties.

Chapter 2

Automotive Applications and Sensor Systems

The increasing complexity of automotive systems require more information for making decisions and therefore, vehicles are equipped with a large number of sensors, some of which are observing the surrounding environment and some providing information about the vehicle itself or the driver. This section provides an introduction to the sensors considered in this thesis and an overview of some potential applications.

2.1 Advanced driver assistance systems

There is a large number of systems in the category of automotive systems referred to as advanced driver assistance systems. In this section we briefly discuss a few of these with the purpose of providing the reader with a context for the contributions of this thesis.

2.1.1 Automotive safety systems

In order to avoid accidents, or to mitigate the consequences caused by an accident, there are systems designed to detect and sometimes also avoid potentially dangerous situations [1, 2]. Some examples of such systems are:

Lane keeping assist/lane departure warning: To avoid unintended lane departures, there are systems that warn the driver [3] and systems that actively steer back into the lane. For these systems to operate, they require knowledge regarding the position and the shape of the lane markings. This information is used to predict if the vehicle will make an unintentional lane change and to calculate the required torque for steering back into the lane.

Collision warning/avoidance/mitigation: There are many systems designed to warn or intervene in situations where a collision is about to happen [4,5]. For example, using a forward-looking radar, it is possible to detect if the vehicle approaches another vehicle at a too high speed. If this is the case, the system can warn the driver or intervene by braking. To avoid false alarms, such as automatically brake when the driver intend to overtake the vehicle ahead, some systems only intervene when it is already too late to avoid a collision and by that only mitigate the damages.

Pedestrian detection: Using cameras, there are systems that identify pedestrians [4,6,7] and if someone approaches the road in front of the vehicle the driver is warned and if no action is taken, the system automatically applies the brakes. Similar systems have been developed in order to avoid accidents involving bicycles or wild animals [8].

An important aspect of automotive safety systems is driver acceptance. For example, if a system intervenes when the driver has full control over the situation, or if it provides many false warnings, the system is perceived as unreliable and annoying and will eventually be turned off. To make these systems more efficient, they can be given access to information about the driver. An attentive driver can be given full control over the situation, while if the driver is tired or distracted, the system can warn or intervene earlier without the driver perceiving it as a false alarm.

2.1.2 Towards self-driving vehicles

When discussing automated vehicles, it is easy to only think about fully autonomous vehicles where the "driver" simply enters a destination and lean back while the vehicle plans and controls the entire trip. However, the level of automation can range from that of a vehicle equipped with an adaptive cruise control or an automatic parking system, to fully autonomous, or self-driving, vehicles [9, 10]. The automation level considers different aspects, such as if a vehicle function includes execution of steering and/or acceleration/deceleration, monitoring of the driving environment and the expectation of the driver to take control. One categorization of automation levels is developed by the US National Highway Traffic Safety Administration (NHTSA) [9], and is summarized as:

- **Level 0 - No automation:** The driver is in control of steering, brake and throttle at all times. The vehicle can be equipped with warning systems, such as blind spot monitoring etc.
- **Level 1 - Function-specific automation:** On this level, the vehicle automation involves systems that can be in limited control of a specific task, such as adaptive cruise control or automatic braking, however not in combination with steering.
- **Level 2 - Combined function automation:** At least two functions working together, such as ACC in combination with a lane following system. Enables both hands-off-wheel and foot-off-pedal operation but where the driver is expected to be available for control at all times and at short notice.
- **Level 3 - Limited self-driving automation:** The vehicle controls all functions under certain traffic conditions and monitor the situation in order to alert the driver when a transition to driver control is required.
- **Level 4 - Full self-driving automation:** The vehicle is designed to be in control of all functions during an entire trip based on a navigation input.

When shifting focus from safety systems to self-driving vehicles, new problems arise. One major difference is that many safety systems are designed for a specific task or scenario, such as lane keeping or forward collision warning. A self-driving vehicle needs to handle more complex scenarios including different situations that arise during normal driving but also more unusual situations such as foreign objects on the road or if the traffic is re-directed due to road construction or due to an accident. In addition, a safety system is only designed to support the driver and can hence be made rather conservative in order to avoid false warnings or interventions. When the objective is to develop self-driving vehicles, the requirement on avoiding false interventions is complemented by strict requirements regarding missed interventions. While considering these demands, the self-driving vehicle shall perform tasks ranging from perceiving the environment and assessment of the traffic situation to self-localization, path planning and vehicle control.

Defense Advanced Research Projects Agency (DARPA) urban challenge [11, 12] Around the world, there are many projects on self-driving vehicles. The most well-known is probably the Google Car [13], but many car manufacturers also focus on this topic. For example, in 2013, a German team had a demonstration where a car drove autonomously along a pre-defined 100 km long route using radar and camera sensors [14]. In the Volvo DriveMe project, the aim is to have 100 cars in open traffic by 2017.

2.2 Sensor systems

Depending on the application, the required information for making a robust system differs and thus affects the choice of sensors. For example, parking assistance systems often use ultrasound sensors to measure the distance to other vehicles and objects. This sensor works well in this setting, but to measure the distance to a vehicle on a highway, maybe up to 150 meters, a radar is used. The subsequent sections provide an overview of the sensors considered in the thesis. The focus is on the type of information provided by the different sensors and how it can be used in automotive systems.

2.2.1 Radar

Many automotive systems make use of radar sensors due to its ability to accurately measure the distance, the angle and the relative speed to both moving and stationary objects. Another important strength is the robustness to weather conditions such as rain, fog and snow [15].

Radar is an acronym for RAdio Detection And Ranging and the basic measuring principle relies on the transmission of microwaves that are reflected by other objects and thereafter registered by a receiver [16]. From the transmitted signal power, only a fraction will return to the receiver. The reflected power depends on several parameters, such as the distance to the object, what type of object it is etc. The received signal is processed in order to extract useful information. A common output from an automotive radar unit is a set of detections, where each detection includes the range and angle to a reflecting feature/object as well as the relative radial velocity. In some settings, the signal strength can be accessible.

Automotive radars are often divided into three categories [17], namely short range radar, medium range radar and long range radar with typical fields of view according to Table 2.1. In automotive settings, different types of radars are used depending on the application. Maybe the aim is to monitor the blind spot region, or observe the traffic in order to warn for approaching vehicles at an intersection, so called cross traffic alert. Often, the vehicle is equipped with a set of radar sensors with the purpose to support different systems.

	Short range radar (SRR)	Medium range radar (MRR)	Long range radar (LRR)
Range	30 m	100 m	250 m
Angle	±80°	±40°	±15°

Table 2.1: Typical properties of the field of view for different radar types.

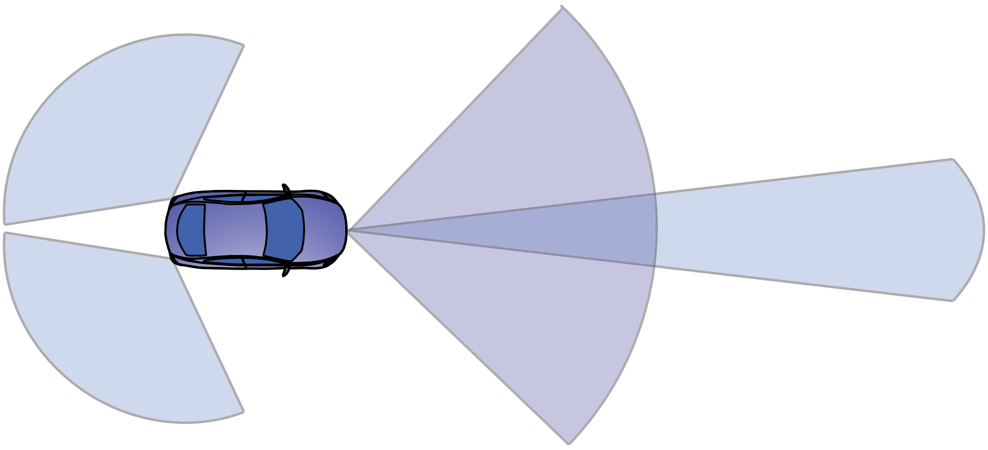


Figure 2.1: An example of a radar setup (not to scale).

2.2.2 Camera

There are several advantages with vision-based sensors [2]. They are cost-efficient sensors that are well suited to detect road and traffic information intended to be clearly visible for the human eye. Depending on the system, either the raw images or a set of extracted features are made available. The main drawbacks with a camera are its sensitivity to varying lighting conditions, such as strong sunlight or poor lighting, as well as heavy rain, fog or dirt.

Forward looking camera: A camera mounted to face forward can be used to monitor the road ahead of the vehicle. By analyzing the captured video frames, it is possible to detect pedestrians, traffic signs, lane markings or other vehicles. Another advantages with a camera is that it enables classification of detected objects.

Driver monitoring: Using one or multiple cameras inside a vehicle enable monitoring of the driver. There are several such monitoring systems on the market and the information they provide differ slightly. Common observations are the position of the head, the head rotation, the gaze direction and eye closure.

2.2.3 Internal sensors

All vehicles are equipped with a large number of internal sensors, with the purpose of monitoring the vehicle and provide the driver, or a system, with information [18]. Internal sensors whose function is regularly used by the driver are the speedometer, the odometer (that measures the travelled distance of a vehicle) and the fuel level indicators. Another example is accelerometers that are used to detect abrupt changes in the velocity of a vehicle. This information can be used for triggering the inflation of airbags in case of an accident.

2.2.4 Global navigation satellite system

Global navigation satellite system (GNSS) is a family of systems that uses signals from satellites in order to locate a receiver on Earth [19]. Like many other technical findings, the development of satellite navigation started within the military area, but over the years it has spread into civil applications. Nowadays, most phones are equipped with a navigation system.

The most well-known system is the global positioning system (GPS), that makes use of 24 satellites in orbit around the Earth. Each satellite transmits its position and the current time while the GPS receiver listens to several satellites and computes its position based on the measured signal propagation time for each satellite. There are many potential sources of errors in this process. For example, the transmitted signals travel through the atmosphere which affects the signal path and speed. Closer to ground the signal can be reflected by buildings, the ground or other objects. This phenomenon is called multipath and affect the position since the signal has travelled a longer distance.

To improve the accuracy in the position measurements, it is possible to use reference stations located at known positions. This way, some of the sources to disturbances, such as atmospheric effects, can be cancelled out. However, it still requires an open area in order to avoid multi-path.

Chapter 3

Bayesian Filtering

In many applications, the aim is to recursively estimate some parameters of interest based on measurements from one or several sensors. It can, for example, be the position and velocity of an aircraft observed by a radar, or the locations of firefighters in a building using inertial sensors mounted on knees and feet. In the Bayesian filtering, the parameters of interest are collected in a state vector, and through filtering of the observations, a posterior distribution of the state is computed. In this chapter, the general filtering problem is defined and the conceptual solution is derived. It is shown that a filtering recursion consists of two steps, namely a prediction step and a measurement update, and it is discussed how these are performed in different types of problems.

3.1 Problem formulation

The general problem can be formulated as finding an estimate of the state vector $\mathbf{x}_k$ using information from a set of sensors. The subscript k denotes the discrete time instant corresponding to continuous time t_k. The available information regarding $\mathbf{x}_k$ is a set of observations, collected by one or several sensors from time 1 to k. The complete set of data is an ordered set denoted by

$$\mathbf{Z}_{1:k} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}, \tag{3.1}$$

where $\mathbf{z}_i$ is a vector containing the observations made at time i. The assumed relations between the state at different times and the received measurements are illustrated in Figure 3.1. From this figure, two key modelling assumptions in filtering can be noticed. Firstly, the evolution of the state vector fulfills the Markov property. That is, a future state only depends on the current

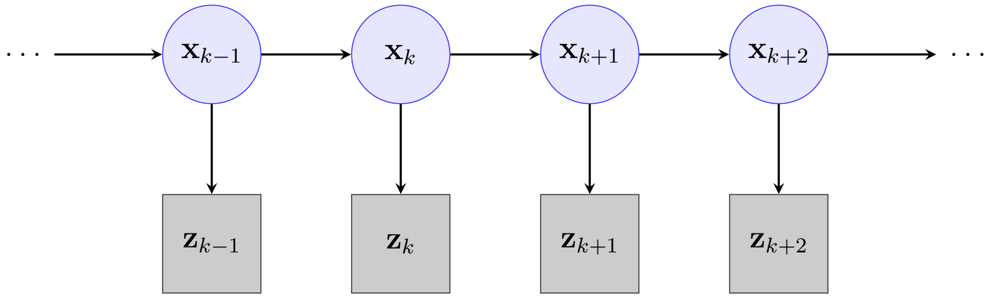


Figure 3.1: The assumed dependencies between the states and the measurements.

state such that:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1},\dots,\mathbf{x}_0) = p(\mathbf{x}_k|\mathbf{x}_{k-1}). \tag{3.2}$$

Secondly, a measurement at time k is only dependent on the current state, $\mathbf{x}_k$, implying that:

$$p(\mathbf{z}_k|\mathbf{x}_k,\dots,\mathbf{x}_0) = p(\mathbf{z}_k|\mathbf{x}_k). \tag{3.3}$$

In a Bayesian filtering setting, the objective is formulated as recursively finding the posterior probability density function (pdf) $p(\mathbf{x}_k|\mathbf{Z}_{1:k})$. This density summarizes the knowledge about the state, given all the data up to and including time k. From the density, an estimate $\hat{\mathbf{x}}_{k|k}$ of the state can be derived according to a chosen optimality criterion. For example, the most likely value of $\mathbf{x}_k$ is referred to as the maximum a posteriori (MAP) estimate and is defined as

$$\hat{\mathbf{x}}_{k|k}^{\text{MAP}} = \arg\max_{\mathbf{x}} p(\mathbf{x}_k|\mathbf{Z}_{1:k}). \tag{3.4}$$

Alternatively, if the aim is to minimize the mean squared error, the estimate can be found as

$$\begin{aligned}\hat{\mathbf{x}}_{k|k}^{\text{MMSE}} &= \arg\min_{\mathbf{x}} \mathbb{E}\Big\{(\mathbf{x}_k-\hat{\mathbf{x}}_k)^T(\mathbf{x}_k-\hat{\mathbf{x}}_k)|\mathbf{Z}_{1:k}\Big\}\\ &= \mathbb{E}\{\mathbf{x}_k|\mathbf{Z}_{1:k}\}\\ &= \int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{Z}_{1:k}) d\mathbf{x}_k. \end{aligned}\tag{3.5}$$

Having established that the posterior density is of primary interest, next the conceptual solution for computing $p(\mathbf{x}_k|\mathbf{Z}_{1:k})$ is detailed.

3.2 Conceptual solution

In this section we derive the equations needed for one filter recursion. More specifically, we show how the posterior from time $k-1$, $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$, and the observations at time k can be used to compute $p(\mathbf{x}_k|\mathbf{Z}_{1:k})$.

By splitting the measurement set into the data from the current time step and from previous times, the posterior probability density function is

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

Then, by using Bayes' law

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{Z}_{1:k-1}) &= \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{Z}_{1:k-1})p(\mathbf{x}_k|\mathbf{Z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{1:k-1})} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{1:k-1})}, \end{aligned} \tag{3.7}$$

where the second equality follows from the assumption in (3.3), i.e., if $\mathbf{x}_k$ is known, the new observations do not depend on previous measurements. In (3.7), the factor $p(\mathbf{z}_k|\mathbf{x}_k)$ is called the likelihood and $p(\mathbf{x}_k|\mathbf{Z}_{1:k-1})$ is the predicted density. The denominator $p(\mathbf{z}_k|\mathbf{Z}_{1:k-1})$ is a normalization factor that ensure that the posterior density integrates to one. Ignoring the normalization, the posterior is found as:

$$\text{Posterior} \propto \text{Prior} \times \text{Likelihood}. \tag{3.8}$$

Consequently, the posterior is found by combining the knowledge in the prior (in this case the predicted density) with the information provided by the new set of measurements.

The predicted distribution is found by marginalizing over the state at the previous time step

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) &= \int p(\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1}) d\mathbf{x}_{k-1} \\ &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Z}_{1:k-1}) p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1}) d\mathbf{x}_{k-1} \\ &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1}) d\mathbf{x}_{k-1}, \end{aligned} \tag{3.9}$$

which is called the Chapman-Kolmogorov equation. It is worth noticing that the last equality follows from the assumption that the state evolution over time satisfies the Markov property. In (3.9), $p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})$ can be identified as the posterior density at time instant $k-1$ and $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the state transition density describing how the state evolves over time.

The state transition density, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, is given by a process (motion) model and the likelihood, $p(\mathbf{z}_k|\mathbf{x}_k)$, is given by a measurement (sensor) model. There models describe the relations between the states and the observations in Figure 3.1 and are commonly expressed as:

$$\begin{aligned} \mathbf{x}_k &= f_{k-1}(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) && (3.10) \\ \mathbf{z}_k &= h_k(\mathbf{x}_k, \mathbf{w}_k), && (3.11) \end{aligned}$$

where $f_{k-1}(\cdot)$ and $h_k(\cdot)$ are potentially non-linear models and where $\mathbf{v}_{k-1}$ and $\mathbf{w}_k$ are process and measurement noise, respectively.

3.3 The Kalman filter

If both the process model and the measurement model are linear with additive Gaussian noise, the models in (3.10) and (3.11) can be formulated as

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_{k-1} && (3.12) \\ \mathbf{z}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{w}_k, && (3.13) \end{aligned}$$

where $\mathbf{v}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ and $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. If, in addition, the prior density of the state, $p(\mathbf{x}_0)$, is Gaussian, the resulting posterior probability density function will also be Gaussian and the filtering can be performed optimally using the Kalman filter [20].

In the Kalman filter, the posterior density at time $k-1$ is given by

$$p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}), \tag{3.14}$$

where $\hat{\mathbf{x}}_{k-1|k-1}$ is the estimates state at time $k-1$ and $\mathbf{P}_{k-1|k-1}$ is the covariance matrix capturing the uncertainties in the estimate. In the prediction step, the posterior in (3.14) is propagated through the process model in order to compute an estimate of $\mathbf{x}_k$ using the data up to and including time $k-1$. Since the Gaussian density is completely described by its mean and covariance, it is sufficient to describe how to compute these quantities. Hence, the mean and covariance of the predicted density are:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} && (3.15) \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}. && (3.16) \end{aligned}$$

In the measurement update step, the predicted state and covariance are updated using the measurement $\mathbf{z}_k$. The result is a Gaussian posterior density, $\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$, whose mean and covariance are computed according to:

$$
\begin{aligned}
\mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k && (3.17)\\
\mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} && (3.18)\\
\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) && (3.19)\\
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T. && (3.20)
\end{aligned}
$$

These steps are iterated in order to find the posterior density at each time instant k.

In many cases, the used models are not linear and the Kalman filter can thus not be applied to the problem. Instead, one must resort to one of many approximate methods, some of which will be briefly presented in the following section.

3.4 Filtering with non-linear models

There are many non-linear models commonly used in object tracking and filtering. One example is the measurement model used when tracking a moving object using range and bearing measurements. Assuming that the state vector contains the position of the object, defined in a Cartesian coordinate frame, $\mathbf{x}_k = [x_k, y_k]^T$. If the sensor is located at the origin, the measurement model is

$$
\begin{bmatrix} r_k \\ \varphi_k \end{bmatrix} = h(\mathbf{x}_k) + \mathbf{w}_k = \begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\ \arctan\left(y_k/x_k\right) \end{bmatrix} + \mathbf{w}_k, \qquad (3.21)
$$

where r_k and φ_k are the measured range and angle, respectively.

In contrast to the linear case, propagating a Gaussian density through this type of non-linear function does not result in a Gaussian density. To perform filtering in this case, there are different approaches. For example, one can linearize the models and apply the Kalman filter equations to the linear state model. Or, the involved densities can be represented by samples that are propagated through the non-linear models. For illustration, we consider the scenario where both the process model and the sensor model are non-linear with additive Gaussian noise. More specifically,

$$
\begin{aligned}
\mathbf{x}_k &= f(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1} && (3.22)\\
\mathbf{z}_k &= h(\mathbf{x}_k) + \mathbf{w}_k, && (3.23)
\end{aligned}
$$

where $\mathbf{v}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ and $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. Next, we present some common filtering methods for problems involving nonlinear models.

3.4.1 The extended Kalman filter

One approach for filtering with nonlinear models is to linearize the models and then use the Kalman filter equations. The linearization in the extended Kalman filter (EKF) [21] is performed by a first order Taylor expansion around the expected value of the state, i.e. for the prediction step at time k the process model is linearized about $\hat{\mathbf{x}}_{k-1|k-1}$. To illustrate the idea in EKF we consider the type of process model in (3.22). Further, we introduce the notation

$$\hat{\mathbf{F}} = \left[\nabla_{\mathbf{x}_{k-1}} f(\mathbf{x}_{k-1})^T\right]^T \Big|_{\mathbf{x}_{k-1}=\hat{\mathbf{x}}_{k-1|k-1}}, \tag{3.24}$$

where $\nabla_{\mathbf{x}_{k-1}} \triangleq \left[\frac{\partial}{\partial \mathbf{x}_{k-1}(1)}, \dots, \frac{\partial}{\partial \mathbf{x}_{k-1}(n)}\right]^T$. Consequently, the first order Taylor expansion of the model at $\hat{\mathbf{x}}_{k-1|k-1}$ is

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1} \\ &\approx f(\hat{\mathbf{x}}_{k-1|k-1}) + \hat{\mathbf{F}}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}) + \mathbf{v}_{k-1} \end{aligned} \tag{3.25}$$

which is linear in $\mathbf{x}_{k-1}$. Based on this linear model, the Kalman filter can be applied to the problem. Using (3.25), the expected value and covariance of $\mathbf{x}_k|\mathbf{Z}_{1:k-1}$ are:

$$\mathbb{E}\{\mathbf{x}_k|\mathbf{Z}_{1:k-1}\} = f(\hat{\mathbf{x}}_{k-1|k-1}) \tag{3.26}$$

$$\mathrm{Cov}(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) = \hat{\mathbf{F}}\mathbf{P}_{k-1|k-1}\hat{\mathbf{F}}^T + \mathbf{Q}_k, \tag{3.27}$$

which can be recognized as the prediction equations in the Kalman filter. In a similar fashion, given the nonlinear sensor model in (3.23), a linearized model can be found by computing the Jacobian

$$\hat{\mathbf{H}} = \left[\nabla_{\mathbf{x}_k} h(\mathbf{x}_k)^T\right]^T \Big|_{\mathbf{x}_k=\hat{\mathbf{x}}_{k|k-1}}. \tag{3.28}$$

This model can then be used in the Kalman filter update equations (3.17) – (3.20).

The EKF is a very popular choice in many applications due to its low complexity and good performance. However, the filter encounters difficulties if the models are highly non-linear or if the linearization point is poor. To deal with these issues, there are modifications to the algorithm. One example is the iterative EKF [22, 23], where an iteration step is included in order to improve the linearization point and consequently improve the estimation.

3.4.2 Sigma point filters

There are several methods in the family of sigma point filters, two of which are named the unscented Kalman filter (UKF) [23,24] and the cubature Kalman filter (CKF) [25]. Similar to the EKF, these are Gaussian filters, meaning that the involved densities are approximated as Gaussians. However, in contrast to the EKF where the process and/or the measurement models are linearized, the sigma point filters work by representing the predicted and posterior densities by a set of weighted deterministic samples, called sigma points. The sigma points are propagated through the non-linear functions and, together with the weights, used to approximate the mean and covariance of the transformed density. The difference between the methods within this family of filters is how the sigma points and the weights are chosen.

Considering the prediction step at time k, a set of N sigma points are chosen to represent the posterior density $p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})$. Denoting the sigma points by $\mathcal{X}^{(i)}$ and their corresponding weights by $W^{(i)}$, the predicted mean and covariance are found as

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &\approx \sum_{i=1}^{N} W^{(i)} f(\mathcal{X}^{(i)}) && (3.29)\\ \mathbf{P}_{k|k-1} &\approx \sum_{i=1}^{N} W^{(i)} \Big(f(\mathcal{X}^{(i)}) - \hat{\mathbf{x}}_{k|k-1}\Big)\Big(f(\mathcal{X}^{(i)}) - \hat{\mathbf{x}}_{k|k-1}\Big)^T. && (3.30)\end{aligned}$$

The measurement update step is performed by constructing a Gaussian approximation to the joint distribution $p(\mathbf{x}_k, \mathbf{z}_k|\mathbf{Z}_{1:k-1})$,

$$\begin{bmatrix}\mathbf{x}_k\\ \mathbf{z}_k\end{bmatrix} \Bigg| \, \mathbf{Z}_{1:k-1} \sim \mathcal{N}\left(\begin{bmatrix}\hat{\mathbf{x}}_{k|k-1}\\ \hat{\mathbf{z}}_{k|k-1}\end{bmatrix}, \begin{bmatrix}\mathbf{P}_{k|k-1} & \mathbf{P}_{\mathrm{xz}}\\ \mathbf{P}_{\mathrm{zx}} & \mathbf{P}_{\mathrm{zz}}\end{bmatrix}\right). \tag{3.31}$$

Then, the conditional distribution is $p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{Z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \hat{\mathbf{P}}_{k|k})$ where the mean and the covariance are given by

$$\begin{aligned}\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{P}_{\mathrm{xz}}\mathbf{P}_{\mathrm{zz}}^{-1}(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}) && (3.32)\\ \hat{\mathbf{P}}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{P}_{\mathrm{xz}}\mathbf{P}_{\mathrm{zz}}^{-1}\mathbf{P}_{\mathrm{zx}}. && (3.33)\end{aligned}$$

The measurement update in a sigma point filter is then carried out by using the sigma points to perform moment matching of the components in (3.31):

$$\hat{\mathbf{z}}_{k|k-1} \approx \sum_{i=1}^{N} W^{(i)} h(\mathcal{X}^{(i)}) \tag{3.34}$$

$$\mathbf{P}_{\mathrm{xz}} \approx \sum_{i=1}^{N} W^{(i)} \Big(\mathcal{X}^{(i)} - \hat{\mathbf{x}}_{k|k-1}\Big)\Big(h(\mathcal{X}^{(i)}) - \mathbf{z}_k\Big)^T \tag{3.35}$$

$$\mathbf{P}_{\mathrm{zz}} \approx \mathbf{R}_k + \sum_{i=1}^{N} W^{(i)} \Big(h(\mathcal{X}^{(i)}) - \mathbf{z}_k\Big)\Big(h(\mathcal{X}^{(i)}) - \mathbf{z}_k\Big)^T. \tag{3.36}$$

3.4.3 Particle filters

In the filtering methods discussed so far, the posterior pdf is Gaussian or approximated as a Gaussian. The accuracy in the Gaussian approximation depends on the models. For example, if the true posterior is multi-modal, the approximation might be very poor. A family of methods that can be used to describe any density is called particle filters [26–28]. The idea in particle filters is to approximate the posterior density by a set of N weighted random samples according to

$$p(\mathbf{x}_k|\mathbf{Z}_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}), \tag{3.37}$$

where $\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(N)}$ are called particles and $w_k^{(1)}, \ldots, w_k^{(N)}$ are their corresponding weights, satisfying $w_k^{(i)} \geq 0$ and $\sum w_k^{(i)} = 1$. Hence, the aim in the particle filter setting is to recursively determine the particle states and the weights.

There are different choices to make in a particle filter, resulting in somewhat different algorithms. However, the basic steps in a particle filter are often the same. First a set of new particles are generated by propagating each particle in time and then the weights are updated using the measurements. A basic particle filter propagates the particles using the process model and updates the weights using the measurement likelihood. Then,

$$\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}) \tag{3.38}$$

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(\mathbf{z}_k|\mathbf{x}_k^{(i)}), \tag{3.39}$$

which implies that particles that describe the measurements well will get larger weights and therefore contribute more to the posterior density in (3.37).

Algorithm 1 Particle filter overview

```
Generate particles, $\mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(N)}$, from a prior distribution $p(\mathbf{x}_0)$.
for $k = 1 : K$ do
    for $i = 1 : N$ do
        Propagate the particle in time, $\mathbf{x}_{k-1}^{(i)} \rightarrow \mathbf{x}_k^{(i)}$
        Update the weight $w_k^{(i)}$ using the measurement vector $\mathbf{z}_k$
    end for
end for
```

Unfortunately, all particle filters degenerate. That is, with time only a few particles will have non-zero weights resulting in a poor approximation of the posterior density in (3.37). To solve this issue, the filters include a resampling step where new particles are generated by sampling from the current approximation of the posterior. Each new sample will have a weight $w_k^{(i)} = 1/N$. Some algorithms, such as the Bootstrap particle filter [26], include resampling in each recursion while other only resample the particles when it is required according to a certain criterion.

Chapter 4

Multiple Object Tracking

In many situations where object tracking is of interest, there are multiple objects to consider and it is also likely that the number of objects in the observed region changes over time. The presence of multiple objects also leads to multiple observations at each time instance, resulting in issues with the measurement-to-track associations. That is, given a set of data, it is not known which object generated which measurement, or even which objects that were detected. In addition, there might be clutter detections among the available measurements. These detections further complicate the difficulties with the unknown associations. All these aspects must be considered when designing a multiple object tracking algorithm.

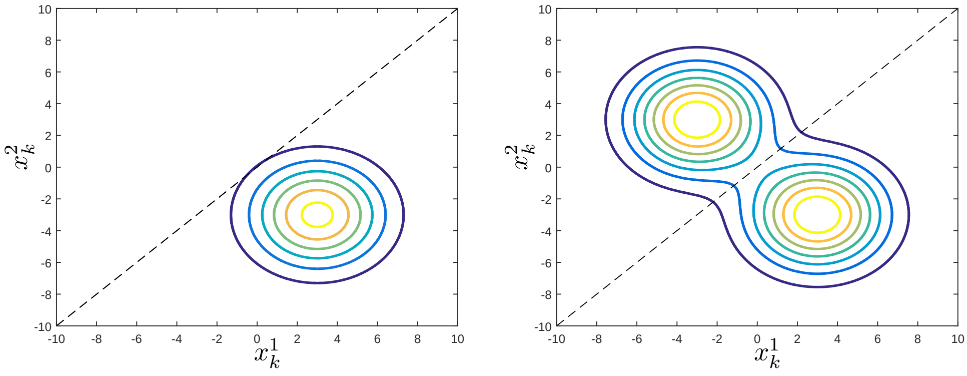
There are two conceptually different approaches to multiple object tracking, namely vector- and set-based methods. To illustrate the difference, we consider a scenario with two objects described by the states $\mathbf{x}_k^1$ and $\mathbf{x}_k^2$, respectively. In the traditional vector-based setting, the multiple object state $\mathbf{x}_k$, is

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_k^1 \\ \mathbf{x}_k^2 \end{bmatrix}. \tag{4.1}$$

That is, the single object states are stacked in order to produce an ordered vector. If instead a set-based approach is adopted, the state variable is an unordered set, X_k, such that

$$X_k = \{\mathbf{x}_k^1, \mathbf{x}_k^2\} = \{\mathbf{x}_k^2, \mathbf{x}_k^1\}. \tag{4.2}$$

Example 4.1 (Vector representation vs set description)
One example of probability density functions describing the objects in the two different settings is illustrated in Figure 4.1. In the vector case, the single peak at $[x_k^1, x_k^2] = [3, -3]$ *indicates that the most likely* x_k^1 *is 3 and the*



(a) The joint density, $p(x_k^1, x_k^2)$, for the vector state.

(b) The set density, $p(\{x_k^1, x_k^2\})$.

Figure 4.1: The probability density functions .

most likely x_k^2 *is -3. Due to the orderless property of the set in* (4.2)*, the set-density has two peaks, one at* $[x_k^1, x_k^2] = [3, -3]$ *and one at* $[x_k^1, x_k^2] = [-3, 3]$*, indicating that one object (either* x_k^1 *or* x_k^2*) is located around 3 and one around -3.*

Both representations can be generalized to an arbitrary number of objects, and depending on the representation, both the problem formulation and the derived solutions will be affected. The next section continues with a discussion on data association for multiple object states as the one in (4.1). In Section 4.2, the theory of tracking using random finite sets, as in (4.2), is introduced. Each theory section is followed by a brief description of practical tracking implementations.

4.1 Data association for a known number of objects

To describe the idea of data association in the vector-based setting, we assume that the number of objects is known. The appearance and disappearance of objects in this setting is not discussed in this thesis.

4.1.1 Problem formulation

Let us consider a scenario with N objects, where each object i is described by a state vector $\mathbf{x}_k^i$. At time k, a sensor delivers a set of measurements denoted $\mathbf{z}_k = [\mathbf{z}_k^1, \ldots, \mathbf{z}_k^{m_k}]$ including both object detections and clutter. The following assumptions are made regarding the objects and the measurements:

- The motion of the objects is independent. Hence, it is enough to describe the single object motion model according to $\mathbf{x}_k^i = f(\mathbf{x}_{k-1}^i) + \mathbf{v}_{k-1}$.
- Each object is assumed to give rise to at most one measurement at each time instant. Due to a probability of detection less than 1, it is possible that the object is not detected at all.
- The measurement set contains both detections from objects and clutter (false alarms). Further, each object-generated measurement is generated by exactly one object.

Theses assumptions imply that the objects are considered point targets and that the sensor has an infinite resolution. In is worth noting that these assumptions have the origin in the traditional application of radar tracking, namely tracking of aircrafts at far distances. In other settings, such as vehicle tracking, the assumptions might not valid and thereby negatively affect the results.

Based one the stated assumptions the objective is to find the posterior probability density function (pdf) of the multiple object state vector $\mathbf{x}_k$. More specifically, the aim is to recursively compute $p(\mathbf{x}_k|\mathbf{Z}_{1:k})$, where, as before, $\mathbf{Z}_{1:k} = \{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$ is the collection of measurements from time 1 to k.

4.1.2 Conceptual solution

Assuming that at time k, there are N objects and a set of m_k observations, we can introduce a data association vector $\mathbf{a}_k = [a_k^1, \ldots, a_k^N]$ where

$$a_k^i = \begin{cases} j & \text{if measurement } j \text{ is assigned to object } i \\ 0 & \text{if object } i \text{ was not detected at time } k. \end{cases} \tag{4.3}$$

Over time, a sequence of association vectors, $\mathbf{A}_{1:k} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]$, can be created and if this sequence is known, the multi object posterior $p(\mathbf{x}|\mathbf{Z}_{1:k}) = p(\mathbf{x}|\mathbf{Z}_{1:k}, \mathbf{A}_{1:k})$ can be found using any of the filtering methods discussed in the previous chapter. However, since the data associations are unknown, for the computation of the posterior pdf to be exact, it must consider all possible hypotheses according to:

$$p(\mathbf{x}_k|\mathbf{Z}_{1:k}) = \sum_t \Pr\left\{\mathbf{A}_{1:k}^t|\mathbf{Z}_{1:k}\right\} p(\mathbf{x}_k|\mathbf{Z}_{1:k}, \mathbf{A}_{1:k}^t). \tag{4.4}$$

However, the number of possible sequences of association vectors grows exponentially over time, making the computation of the exact pdf in (4.4) intractable. As often in practice, this forces us to resort to approximate methods.

4.1.3 Data association methods

In this section, we review some common algorithms used to approximate (4.4). A simple way to perform data association is to adopt the nearest neighbour method. That is, at each time instant, the most likely measurement is assigned to each object and then used to update the object state vector. In this method, each object is handled separately, possibly resulting in that more than one object is updated using the same measurement, an issue that may lead to coalescence of object tracks. As an improvement, in global nearest neighbour, all objects and measurements are considered simultaneously in order to find the best global data association hypothesis. The main benefit with these approaches is that they avoid treating multiple hypotheses. On the other hand, by making a hard decision, the approximation of (4.4) might be poor.

To take multiple hypotheses into account, the data association can be handled by the probabilistic data association (PDA) filter [29] or the joint PDA (JPDA) filter [30]. The relation between PDA and JPDA is similar to that between nearest neighbour and global nearest neighbour, that is, while PDA only considers local hypotheses, JPDA also takes the global hypotheses into account. In both filters, the posterior density for object i is a weighted sum of the posterior densities computed under the different association hypotheses

$$p(\mathbf{x}_k^i|\mathbf{Z}_{1:k}) = \sum_{j=1}^{m_k} p(\mathbf{x}_k^i|\mathbf{Z}_{1:k}, a_k^i = j) \Pr\left\{a_k^i = j|\mathbf{Z}_{1:k}\right\}. \tag{4.5}$$

The difference between PDA and JPDA lies in the computation of the association probabilities $\Pr\{a_k^i = j|\mathbf{Z}_{1:k}\}$. While PDA treats the problem as several single object problems, in JPDA it must be ensured that no association conflicts occur. Assuming that the computation of the posterior densities $p(\mathbf{x}_k^i|\mathbf{Z}_{1:k}, a_k^i = j)$ is performed using a Gaussian filter, the posterior density in (4.5) is a Gaussian mixture. At each time step, this mixture is approximated by a single Gaussian with the same mean and the same covariance as the mixture.

In contrast to the methods discussed so far, which reduces the mixture in (4.4) to a single (Gaussian) density, in multiple hypothesis tracking (MHT) several sequences of possible association hypotheses are propagated to the next time instant [31]. This way, the decision regarding the most probable hypothesis is postponed to a later time when more data have been received. There exists several methods to limit the number of considered association sequences. Two common approaches are pruning of hypotheses with low probabilities, and merging of hypotheses that result in similar state descriptions.

4.2 Random finite sets for filtering

In this section we introduce the concept of random finite sets (RFS) and describe the two most common methods for incorporating them in a Bayesian filtering framework. In the traditional filtering setting, such as in the previous sections, the aim is to estimate a state vector $\mathbf{x}_k$ based on a set of observations. This state vector is stochastic and can contain information about one object or several stacked single object state vectors. However, regardless of how many objects the state vector describes, the number, and thus the length of the vector, is typically assumed to be known. For a random finite set on the other hand, both the number of elements in the set, i.e. the number of objects, as well as their states, are stochastic [32].

Let us consider an RFS $X = \{\mathbf{x}^1, \dots, \mathbf{x}^n\}$. The number of elements in X is called the cardinality and is denoted by $|X|$. The related cardinality distribution is a probability mass function $p(n)$, which gives the probability that the set contains exactly n elements/objects, i.e. $\Pr\{|X| = n\}$. For example, with probability $\Pr\{|X| = 0\}$ the RFS X is the empty set $\emptyset$, with probability $\Pr\{|X| = 2\}$ the set contains two elements such that $X = \{\mathbf{x}^1, \mathbf{x}^2\}$, and so on.

Similar to a random vector, an RFS can be described by a probability density function. This pdf can be defined using the joint pdf of the ordered single object states. That is,

$$p\Big(\{\mathbf{x}^1, \dots, \mathbf{x}^n\} = \{\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_n\}\Big) = \sum_{i=1}^{n!} p\Big(\mathbf{x}^1 = \boldsymbol{\alpha}_{\sigma_i(1)}, \dots, \mathbf{x}^n = \boldsymbol{\alpha}_{\sigma_i(n)}\Big), \tag{4.6}$$

where σ_i denotes the i:th permutation of the index vector $[1, 2, \dots, n]$. In addition, an RFS is associated with an intensity function, $v(\mathbf{x})$, which describes the concentration of objects over the single object state space. More specifically, the intensity function is defined by the property:

$$\int_S v(\mathbf{x}) d\mathbf{x} \quad = \quad \mathbb{E}\{|X \cap S|\}, \tag{4.7}$$

where S is a region in the single-target state space. Hence, the integral in (4.7) describes the expected number of elements/objects in X that are in the region S. Similarly, $\int v(\mathbf{x})d\mathbf{x}$ is the expected number of objects in X.

Before discussing the problem formulation and the most common solutions within the set-based tracking framework, we present two central types of random finite sets and illustrate their differences by an example.

Cluster RFS

A cluster RFS is defined as a random finite set whose elements are independent and identically distributed (i.i.d.) with an arbitrary cardinality distribution $p(n)$. The set density for such an RFS is given as

$$p(\{\mathbf{x}^1, \ldots, \mathbf{x}^n\}) = n!p(n)p(\mathbf{x}^1)\cdots p(\mathbf{x}^n), \tag{4.8}$$

where $p(\mathbf{x}^i) = v(\mathbf{x}^i)/\int v(\mathbf{x}^i)d\mathbf{x}^i$ is the single object probability density function. The factor $n!$ is related to the fact that there are $n!$ vectors that correspond to the same set.

Poisson RFS

A random finite set, $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$, is called a Poisson RFS if its elements are i.i.d. and the number of elements in the set is Poisson distributed, i.e., $p(n) = \lambda^n \exp(-\lambda)/n!$ where $\lambda = \int v(\mathbf{x})d\mathbf{x}$ is the expected number of elements in X. Inserting this cardinality distribution in (4.8), the resulting set density is

$$\begin{aligned} p(\{\mathbf{x}^1, \ldots, \mathbf{x}^n\}) &= \lambda^n e^{-\lambda} p(\mathbf{x}^1)\cdots p(\mathbf{x}^n) \\ &= e^{-\lambda} v(\mathbf{x}^1)\cdots v(\mathbf{x}^n) \\ &= \exp\Big(-\int v(\mathbf{x})d\mathbf{x}\Big) v(\mathbf{x}^1)\cdots v(\mathbf{x}^n). \end{aligned} \tag{4.9}$$

That is, similar to a Poisson distribution that is completely characterized by its mean, a Poisson RFS can be described by its intensity function, $v(\mathbf{x})$.

Example 4.2 (Cluster and Poisson RFS)
To illustrate the properties of the Poisson and the cluster RFS, we study a scalar problem, that is, the single object state is a scalar denoted by x. The intensity function of the RFS is depicted in Figure 4.2.

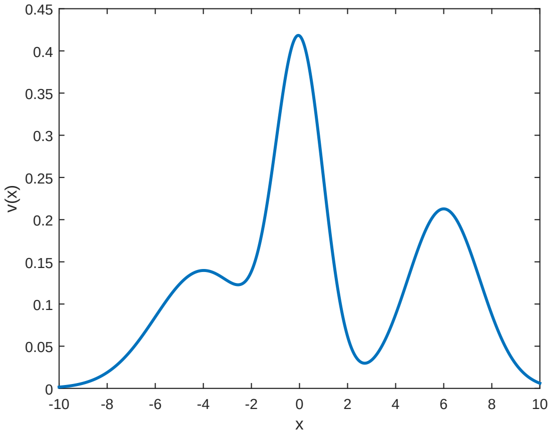


Figure 4.2: An example of an intensity function in one dimension.

Considering a Poisson RFS, the expected number of objects is found as $\hat{N} = \int v(x)dx = 2.5$ *implying that the cardinality distribution for this RFS is* $p(n) = \lambda^n e^{-\lambda}/n!$ *with* $\lambda = 2.5$ *as illustrated in Figure 4.3.*

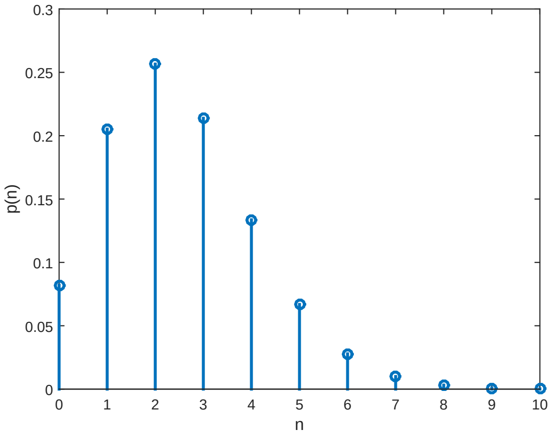
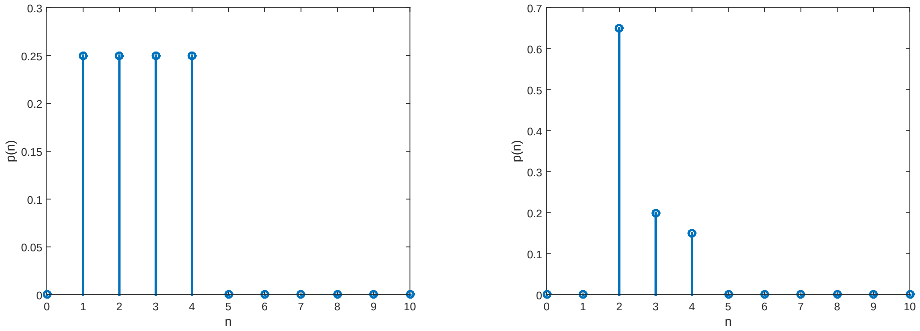


Figure 4.3: The cardinality distribution for the Poisson RFS, i.e. a Poisson distribution with $\lambda = 2.5$.

For a cluster RFS with the same intensity function, the expected number of objects is still $\hat{N} = \int v(x)dx = 2.5$*. However, the intensity function is not enough to define the cardinality distribution. Instead, the cardinality distribution can be any probability mass function with a mean that equals* $\hat{N} = \sum np(n) = \int v(x)dx$*. Two examples of cardinality distributions that fulfill this criterion are shown in Figure 4.4.*



(a) One of many possible cardinality distributions.

(b) Another of many possible cardinality distributions.

Figure 4.4: An illustration of the properties of a cluster RFS.

4.2.1 Problem formulation

At the discrete time instance k, we denote the number of objects by n_k, and their individual states by $\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^{n_k}$, which take values in the single-object state space. The complete multi-object state at time k is defined as the unordered set:

$$X_k = \{\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^{n_k}\}. \tag{4.10}$$

Since the number of objects as well as their states are random variables, X_k is a random finite set. In a similar fashion, at time k, the used sensors provide a set of measurements:

$$Z_k = \{\mathbf{z}_k^1, \mathbf{z}_k^2, \ldots, \mathbf{z}_k^{m_k}\}. \tag{4.11}$$

This set consists of both object-generated measurements and clutter, and it is unknown which measurements originate from the objects and which are clutter. The objective in the multi-object tracking setting is to recursively compute the posterior set density $p(X_k|Z_{1:k})$ where, similar as in previous sections, $Z_{1:k}$ denotes the collection of all data from time 1 to k.

4.2.2 Conceptual solution

When representing the objects and the measurements as sets, there is a need for an alternative Bayesian filter that is able to handle set densities. In [32], the multi-object Bayesian filter is given by the equations

$$p(X_k|Z_{1:k-1}) = \int p(X_k|X_{k-1})p(X_{k-1}|Z_{1:k-1})\delta X_{k-1} \tag{4.12}$$

$$p(X_k|Z_{1:k}) = \frac{p(Z_k|X_k)p(X_k|Z_{1:k-1})}{\int p(Z_k|X_k)p(X_k|Z_{1:k-1})\delta X_k}, \tag{4.13}$$

where the integrals are set integrals defined as,

$$\int p(X)\delta X = p(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int p(\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}) d\mathbf{x}_1 \cdots d\mathbf{x}_n. \tag{4.14}$$

This integral considers the variability in the number of elements in the set X, as well as the distribution of the elements.

In the remainder of this chapter, we discuss the two most common multi-object Bayesian filters, namely the probability hypothesis density (PHD) filter and the cardinalized PHD (CPHD) filter.

4.2.3 The probability hypothesis density filter

In the PHD filter the intensity function is propagated over time, and each filter recursion consists of a prediction and a measurement update of the intensity function. The filter relies on a set of assumptions and approximations, most of which are common in tracking theory:

- The predicted multi-target RFS is a Poisson process, i.e. the targets are assumed to be independent and identically distributed (i.i.d.) with a Poisson cardinality distribution. This assumption is specific for the PHD filter and, as will be discussed later, it contributes to the main drawbacks with the filter.

- Each target evolves and generates measurements independently of all other targets.

- The birth of new targets and the survival of existing targets are independent of each other.

- New targets can appear through spontaneous birth or through spawning from existing targets.

- The clutter RFS is a Poisson process

- The clutter measurements are independent of the target-generated measurements.

Under these assumptions, and using finite set statistics, the PHD filter prediction and measurement update equations were derived in [33]. A major advantage with the PHD filter is that it avoids explicit treatment of the data associations hypotheses. In addition, the filter naturally incorporates the appearance and disappearance of objects. To describe appearing objects, the process model includes a birth process that allows new objects to appear, both spontaneously and through spawning from existing objects. In a similar manner, disappearing objects are treated by a survival/death process.

As mentioned, the assumption that the predicted objects RFS is a Poisson process leads to weaknesses of the filter. Firstly, the cardinality estimates are associated with a high variance, i.e. large uncertainties, something that makes the filter very sensitive to missed detections. In [34], this property is clearly illustrated. Secondly, identically distributed objects is a poor assumption in many problems. This is further discussed in Example 4.3.

Example 4.3 (Independent and identically distributed objects)
Considering a scenario with two scalar objects that are identically distributed, their single-object densities are both $p(x)$. Here, $p(x)$ is a sum of two Gaussians according to:

$$p(x) = w_1 \mathcal{N}(x; m_1, \sigma_1^2) + w_2 \mathcal{N}(x; m_2, \sigma_2^2), \tag{4.15}$$

where $m_1 = -5$, $m_2 = 5$, $w_1 = w_2 = 0.5$ and $\sigma_1^2 = \sigma_2^2 = 1$. As depicted in Figure 4.5, $p(x)$ has two well-separated peaks. Having two independent objects described by this density, the probability that both objects are described by the first component is 0.25. Similarly, with probability 0.25, both objects are at the second component. Consequently, even if the data clearly indicates that there are two objects, the probability of having one object at each Gaussian is only 0.5.

4.2.4 The cardinalized probability hypothesis density filter

To overcome the issues associated with the cardinality estimation in the PHD filter, the CPHD filter was derived [35]. The assumptions adopted in the CPHD filter are similar to that of the PHD filter, but differs at some points. Firstly, the original CPHD filter does not incorporate object spawning. Secondly, the most important difference is that while the cardinality distribution of the object RFS is approximated as Poisson in the PHD filter, it is arbitrary in the CPHD filter.

As discussed in Example 4.2, the intensity function of a Poisson RFS contains enough information to specify the cardinality distribution. However,

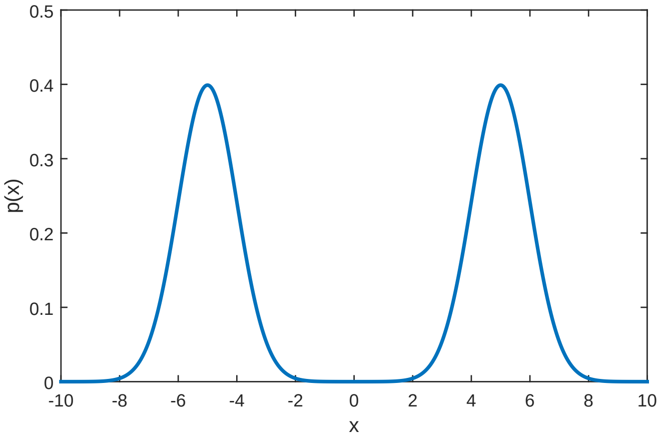


Figure 4.5: The single-object density in Example 4.3.

this is not the case for a cluster RFS. Consequently, while it is sufficient to use intensity function in PHD filter, the CPHD propagates both the intensity function, $v(\mathbf{x})$, and the cardinality distribution, $p(n)$, over time. Thus, each iteration consists of a prediction and a measurement update of both the intensity and the cardinality.

4.2.5 Extensions to the PHD and the CPHD filters

During the years since the PHD and the CPHD filters were first presented, a lot of research has been carried out in order to make the filters practical and to adapt them to different tracking problems. In this section, we provide an overview of some of these results.

Gaussian mixture implementations

In order to make the PHD and the CPHD filters useful, it is necessary to have a representation of the intensity function that is practical. One representation that has been a part of the breakthrough of the filters is the Gaussian mixture versions of PHD [36] and CPHD [37]. In this setting, it is assumed that the intensity function $v(\mathbf{x})$ can be described as a weighted sum of Gaussians

$$v_{k|k}(\mathbf{x}) = \sum_{j=0}^{J_{k|k}} w_{k|k}^{(j)} \mathcal{N}(\mathbf{x}; \mathbf{m}_{k|k}^{(j)}, \mathbf{P}_{k|k}^{(j)}), \tag{4.16}$$

where $w_{k|k}^{(j)}$, $\mathbf{m}_{k|k}^{(j)}$ and $\mathbf{P}_{k|k}^{(j)}$ are the weight, mean and covariance of the j:th component. As a result, each Gaussian can be handled in a similar way as in the Kalman filter framework. The main drawback is that the number of components in the mixture grows over time and it is necessary to employ a method for limiting the number of Gaussians. Common approaches are merging components that are similar according to some criterion, and pruning components with weights lower than a certain threshold.

Applications

The popularity of the PHD and CPHD filter is shown in the literature, where the filters have been applied and adapted to many problems, for example:

- **Extended objects tracking.** Both filters have been derived for extended object tracking [38–40]. That is, these version of the filters are capable of jointly estimating the position, velocity and extension of the object.
- **Mapping** [41,42].
- **Simultaneous localization and mapping (SLAM).** In SLAM, the aim is to simultaneously estimate the ego position and a map of the observed surroundings. In this setting, the PHD filter has been applied for describing the map [43].
- **Unknown clutter intensity** [44].

In addition, the evaluation of a tracking algorithm when the number of objects is unknown, requires a metric that takes both the number of objects as well as the state estimates into account. For this purpose, the optimal sub-pattern assignment (OSPA) was proposed [45].

Chapter 5

Parameter and function estimation

When the aim is to estimate constant parameters or functions there are many well-suited methods (both Bayesian and non-Bayesian) to choose from. The focus in this chapter is on three such methods used in this thesis. First, it is shown how the Expectation Maximization (EM) algorithm can be used to find either a maximum likelihood (ML) estimate or a maximum a posteriori (MAP) estimate of a parameter vector. Second, using Variational Bayesian EM, it is discussed how an inference problem can be re-formulated as an optimization problem with the objective of finding distributions of the parameters of interest. Third, we introduce the Gaussian process, a non-parametric Bayesian method that can be used to learn unknown functions from data.

5.1 Expectation maximization

The expectation maximization (EM) algorithm [46] is an efficient method for parameter estimation in problems that include hidden (latent) variables. These variables are not observed but may be important to explain the relation between parameters of interest and the observations. In many automotive and tracking applications, the hidden variables correspond to the unknown measurement to object associations [47]. In the most common setting, the EM algorithm is used to find the maximum likelihood estimate of some parameters $\boldsymbol{\theta}$, given some data $\mathbf{X}$. However, if prior knowledge regarding the parameters is available, it is straightforward to adjust the algorithm to instead compute the maximum a posteriori estimate. In both cases, the estimation is performed iteratively where, as suggested by the name, each iteration consists of an expectation step and a maximization step.

5.1.1 Maximum likelihood EM

In the maximum likelihood EM (ML-EM) algorithm, the aim is to find the parameter vector, $\boldsymbol{\theta}$, that maximizes the likelihood $p(\mathbf{X}|\boldsymbol{\theta})$. This is equivalent to maximizing the log-likelihood according to

$$\boldsymbol{\theta}^{ML} = \arg\max_{\boldsymbol{\theta}} \ \log p(\mathbf{X}\big|\boldsymbol{\theta}). \tag{5.1}$$

Considering models that are formulated using hidden variables, $\mathbf{Z}$, the likelihood can be expressed through marginalization over $\mathbf{Z}$:

$$\log p(\mathbf{X}|\boldsymbol{\theta}) = \log \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}). \tag{5.2}$$

Since EM is an iterative solution, the objective is to find a method that increases the log-likelihood as much as possible in each iteration [48]. Consequently, the aim is to maximize $\log p(\mathbf{X}|\boldsymbol{\theta}^{(t+1)}) - \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)})$, where the superscript (t) indicates the estimate at iteration t. The increase in the log-likelihood can be expressed as:

$$\begin{aligned}\log p(\mathbf{X}|\boldsymbol{\theta}^{(t+1)}) - \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}) &= \log \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^{(t+1)}) - \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}) \\ &= \log \sum_{\mathbf{Z}} \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^{(t+1)})}{p(\mathbf{X}|\boldsymbol{\theta}^{(t)})} \\ &= \log \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^{(t+1)})}{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^{(t)})}. \qquad (5.3)\end{aligned}$$

Working on an expression that includes the logarithm of a sum is not practical. Using Jensen's inequality [49], (5.3) can be rewritten as

$$\log p(\mathbf{X}|\boldsymbol{\theta}^{(t+1)}) \geq \log p(\mathbf{X}|\boldsymbol{\theta}^{(t)}) + \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \log \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^{(t+1)})}{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^{(t)})}, \tag{5.4}$$

which provides a lower bound on the log-likelihood. The maximization of the likelihood can thus be interpreted as the maximization of the lower bound.

By ignoring the terms in (5.4) that are constant with respect to $\boldsymbol{\theta}^{(t+1)}$, we find the new estimate of $\boldsymbol{\theta}$ as

$$\begin{aligned}\boldsymbol{\theta}^{(t+1)} &= \arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}) \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{Z}}\{\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})|\mathbf{X}, \boldsymbol{\theta}^{(t)}\}. \qquad (5.5)\end{aligned}$$

In (5.5), the two steps of the algorithm become clear:

Expectation-step: Conditioned on the current parameter estimate, $\boldsymbol{\theta}^{(t)}$, the expectation of $\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$, with respect to the hidden variable, is computed.

Maximization-step: A new parameter estimate, $\boldsymbol{\theta}^{(t+1)}$, is found by maximization of $\mathbb{E}_{\mathbf{Z}}\{\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})|\mathbf{X}, \boldsymbol{\theta}^{(t)}\}$.

Starting with an initial parameter vector $\boldsymbol{\theta}^{(0)}$, these two steps are iterated until a chosen convergence criterion is fulfilled. However, it is worth noting that a drawback with the EM algorithm is its sensitivity to the choice of initialization point. An unfavorable initialization may cause the algorithm to converge to a local optima.

Example 5.1 (EM for Gaussian mixture estimation)
In this example we have a set of data denoted by $\mathbf{X} = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$, where $N = 500$ and each observation is a 2-dimensional vector. This data set is illustrated in Figure 5.1. Suppose that each measurement $\mathbf{x}^i$ is an observation of a Gaussian mixture with three components. That is,

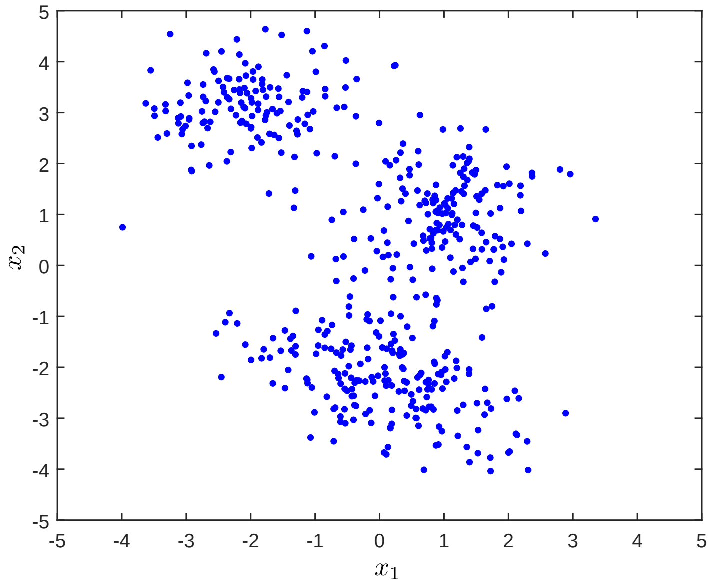


Figure 5.1: Illustration of the data set $\mathbf{X} = [\mathbf{x}^1, \dots, \mathbf{x}^N]$ where $N = 500$ and each $\mathbf{x}^i = [x_1^i, x_2^i]^T$.

$$p\Big(\mathbf{x}^i \Big| w_1, \boldsymbol{\mu}_1, \mathbf{P}_1, w_2, \boldsymbol{\mu}_2, \mathbf{P}_2, w_3, \boldsymbol{\mu}_3, \mathbf{P}_3\Big) = \sum_{j=1}^{3} w_j \mathcal{N}\Big(\mathbf{x}^i; \boldsymbol{\mu}_j, \mathbf{P}_j\Big). \tag{5.6}$$

Introducing the parameter vector $\boldsymbol{\theta} = [w_1, \boldsymbol{\mu}_1, \mathbf{P}_1, w_2, \boldsymbol{\mu}_2, \mathbf{P}_2, w_3, \boldsymbol{\mu}_3, \mathbf{P}_3]$*, the objective is to compute the ML estimate of* $\boldsymbol{\theta}$ *based on the data* $\mathbf{X}$*. The main difficulty in this problem is that the data associations are unknown. That is, it is not know from which Gaussian component that each observation originates. Instead, the associations are introduced as hidden variables, denoted by* $\mathbf{Z}$*.*

As shown in (5.5)*, the EM algorithm starts with the log-likelihood:*

$$\log p(\mathbf{X}|\mathbf{Z},\boldsymbol{\theta}) = \log p(\mathbf{X}|\mathbf{Z},\boldsymbol{\theta}) + \log Pr\left\{\mathbf{Z}|\boldsymbol{\theta}\right\}. \tag{5.7}$$

In this example, $\mathbf{Z}$ *is a matrix whose elements* z_{ij} *are 1 if measurement* i *is associated with component* j*, and 0 otherwise. Since each observation originates from one of the Gaussian components,* $\mathbf{Z}$ *must have exactly one non-zero element in each row. If this is not the case,* $Pr\left\{\mathbf{Z}|\boldsymbol{\theta}\right\} = 0$ *implying that* $p(\mathbf{X}|\mathbf{Z},\boldsymbol{\theta}) = p(\mathbf{X}|\mathbf{Z},\boldsymbol{\theta})Pr\left\{\mathbf{Z}|\boldsymbol{\theta}\right\} = 0$*. Hence, in the following discussion we only consider* $\mathbf{Z}$ *that fulfill the above requirement, resulting in:*

$$p(\mathbf{X}|\mathbf{Z},\boldsymbol{\theta}) = \prod_{i=1}^{N}\prod_{j=1}^{3}\Big(\mathcal{N}(\mathbf{x}^i;\boldsymbol{\mu}_j,\mathbf{P}_j)\Big)^{z_{ij}} \tag{5.8}$$

$$p(\mathbf{Z}|\boldsymbol{\theta}) = \prod_{i=1}^{N}\prod_{j=1}^{3}\left(\omega_j\right)^{z_{ij}}. \tag{5.9}$$

From these expressions we can formulate the log-likelihood:

$$\log p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta}) = \sum_{i=1}^{N}\sum_{j=1}^{3} z_{ij}\Big(\log\omega_j - \log(2\pi) - \frac{1}{2}\log|\mathbf{P}_j| - \frac{1}{2}(\mathbf{x}^i-\boldsymbol{\mu}_j)^T\mathbf{P}_j^{-1}(\mathbf{x}^i-\boldsymbol{\mu}_j)\Big). \tag{5.10}$$

Computing the expectation of (5.10) *with respect to* $\mathbf{Z}$ *is consequently equivalent to computing the expectation of* z_{ij}*. At iteration* $t+1$*, this expectation is found according to*

$$\mathbb{E}\Big\{z_{ij}\big|\mathbf{X},\boldsymbol{\theta}^{(t)}\Big\} = Pr\Big\{z_{ij}=1\big|\mathbf{X},\boldsymbol{\theta}^{(t)}\Big\} = \omega_j^{(t)}. \tag{5.11}$$

In the M step, we find a new parameter vector $\boldsymbol{\theta}^{(t+1)}$*. Since the log-likelihood is a sum of simple terms, the maximization can be performed by solving:*

$$\frac{\partial}{\partial\theta_j}\mathbb{E}_{\mathbf{Z}}\Big\{\log p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})\Big|\mathbf{X},\boldsymbol{\theta}^{(t)}\Big\} = 0, \tag{5.12}$$

for each parameter θ_j *in* $\boldsymbol{\theta}$*. This step results in a new estimate,* $\boldsymbol{\theta}^{(t+1)}$*, that is used in the E step in the next iteration.*

5.1.2 Maximum a posteriori EM

Based on the discussion regarding the maximum likelihood EM algorithm, it is straightforward to derive an EM solution for MAP estimation. The problem of finding the parameters that maximize the posterior density, or equivalently the logarithm of the posterior, is closely related to the ML setting. From Bayes' law, it follows that:

$$\log p(\boldsymbol{\theta}|\mathbf{X}) = \log p(\mathbf{X}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log p(\mathbf{X}), \tag{5.13}$$

where $p(\boldsymbol{\theta})$ captures the prior knowledge of the parameters. Ignoring the term that is constant with respect to $\boldsymbol{\theta}$, the maximization can be expressed as

$$\begin{aligned} \boldsymbol{\theta}^{\mathrm{MAP}} &= \arg\max_{\boldsymbol{\theta}} \; \log p(\mathbf{X}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \; \mathbb{E}_{\mathbf{Z}}\{\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})|\mathbf{X}, \boldsymbol{\theta}^{(t)}\} + \log p(\boldsymbol{\theta}). \end{aligned} \tag{5.14}$$

Hence, to find the MAP estimate, the expectation step is identical to the ML case while the maximization step also takes the prior knowledge regarding the parameters into account.

5.2 Variational Bayesian EM

In the variational Bayesian EM (VBEM) [50, 51], no distinction is made between the parameters, $\boldsymbol{\theta}$, and the hidden variables, $\mathbf{Z}$. Instead, both $\boldsymbol{\theta}$ and $\mathbf{Z}$ are treated as random variables described by their respective distributions.

To emphasize the equal treatment of all variables, the considered parameters are here denoted $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. The problem is hence to find the joint posterior density $p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathbf{X})$. However, in many problems the joint posterior is hard to find and instead, in VBEM, the posterior is approximated by a factorization

$$p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathbf{X}) \approx q_1(\boldsymbol{\theta}_1)q_2(\boldsymbol{\theta}_2). \tag{5.15}$$

Employing this approximation, the aim is to find $q_1(\boldsymbol{\theta}_1)$ and $q_2(\boldsymbol{\theta}_2)$ that make the approximation in (5.15) optimal in the "exclusive" Kullback-Leibler (KL) sense [52, 53]. Consequently, in VBEM, the considered inference problem is reformulated into an optimization problem where the aim is to minimize

$$\mathrm{KL}(q_1(\boldsymbol{\theta}_1)q_2(\boldsymbol{\theta}_2)||p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathbf{X})) = \int\int q_1(\boldsymbol{\theta}_1)q_2(\boldsymbol{\theta}_2) \log \frac{q_1(\boldsymbol{\theta}_1)q_2(\boldsymbol{\theta}_2)}{p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathbf{X})} d\boldsymbol{\theta}_1 d\boldsymbol{\theta}_2 \tag{5.16}$$

with respect to the distributions $q_1(\boldsymbol{\theta}_1)$ and $q_2(\boldsymbol{\theta}_2)$.

A common technique to perform this optimization is to iteratively minimize the KL divergence with respect to one of the distributions while holding the other one fixed. We start the derivation by keeping $q_2(\boldsymbol{\theta}_2)$ fixed and consider the KL divergence as a function of $q_1(\boldsymbol{\theta}_1)$. That is,

$$\begin{aligned}\mathrm{KL}(\boldsymbol{\theta}_1) &= \int\int q_1(\boldsymbol{\theta}_1)q_2(\boldsymbol{\theta}_2)\Big[\log q_1(\boldsymbol{\theta}_1) + \log q_2(\boldsymbol{\theta}_2) - \log p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathbf{X})\Big]d\boldsymbol{\theta}_1 d\boldsymbol{\theta}_2 \\ &= \int q_1(\boldsymbol{\theta}_1)\Big[\log q_1(\boldsymbol{\theta}_1) - \int q_2(\boldsymbol{\theta}_2)\log p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathbf{X})d\boldsymbol{\theta}_2\Big]d\boldsymbol{\theta}_1 + C_1,\end{aligned} \tag{5.17}$$

where the constant C_1 contains all terms that do not depend on $\boldsymbol{\theta}_1$. We continue by introducing a density

$$g(\boldsymbol{\theta}_1) \propto \exp\left(\int q_2(\boldsymbol{\theta}_2)\log p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathbf{X})d\boldsymbol{\theta}_2\right), \tag{5.18}$$

such that $\log g(\boldsymbol{\theta}_1) = \log C_2 + \int q_2(\boldsymbol{\theta}_2)\log p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathbf{X}))d\boldsymbol{\theta}_2$. Using the density $g(\boldsymbol{\theta}_1)$, we can rewrite (5.17) as

$$\begin{aligned}\mathrm{KL}(\boldsymbol{\theta}_1) &= \int q_1(\boldsymbol{\theta}_1)\Big[\log q_1(\boldsymbol{\theta}_1) - \log g(\boldsymbol{\theta}_1)\Big]d\boldsymbol{\theta}_1 + C_1 + \log C_2 \\ &= \int q_1(\boldsymbol{\theta}_1)\frac{\log q_1(\boldsymbol{\theta}_1)}{\log g(\boldsymbol{\theta}_1)}d\boldsymbol{\theta}_1 + C \\ &= \mathrm{KL}(q_1(\boldsymbol{\theta}_1)||g(\boldsymbol{\theta}_1)) + C.\end{aligned} \tag{5.19}$$

This expression is minimized when $q_1(\boldsymbol{\theta}_1) = g(\boldsymbol{\theta}_1)$ implying that

$$\begin{aligned}q(\boldsymbol{\theta}_1) &\propto \exp\left(\int q_2(\boldsymbol{\theta}_2)\log p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathbf{X})d\boldsymbol{\theta}_2\right) \\ &= \exp\Big(\mathbb{E}_{q(\theta_2)}\{\log p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathbf{X})\}\Big).\end{aligned} \tag{5.20}$$

Similarly, the optimal $q_2(\boldsymbol{\theta}_2)$, given $q_1(\boldsymbol{\theta}_1)$, is

$$q(\boldsymbol{\theta}_2) \propto \exp\Big(\mathbb{E}_{q(\theta_1)}\{\log p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathbf{X})\}\Big). \tag{5.21}$$

For many problems, (5.20) and (5.21) have closed form solutions. More specifically, this is the case for models in the conjugate-exponential family, i.e. models with an exponential likelihood and a prior that is conjugate to that likelihood [54]. This family include many common models involving Normal, Poisson, Gamma, Wishart and Inverse Wishart distributions. In case the used model is not in the conjugate-exponential family, there is a need for further approximations [55].

A drawback with VBEM is that, even for fairly simple models, the derivations required for the analytical approximation of the posterior density can be troublesome. However, the resulting VBEM algorithm is computationally efficient compared to many alternative methods that rely on Markov Chain Monte Carlo sampling [56].

5.3 Gaussian processes

With the focus on learning unknown functions based on data, in this section we introduce the concept of Gaussian processes [57, 58]. This is a Bayesian non-parametric method that can be used to describe a distribution over functions. A major advantage with this approach is that Gaussian processes are very flexible and are capable to describe very complex functions. The following discussion is based on the definition of a Gaussian process:

Definition: *"A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution."*

For illustration, we consider an unknown function f, with argument $\mathbf{x} \in \mathbb{R}^n$ and scalar output y. In the discussion on Bayesian filtering, many of the involved densities are assumed to be Gaussian, characterized by its mean and covariance. Similarly, a prior for the function f can be defined by a Gaussian process that is specified by its mean function and covariance function. This prior is denoted by

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big), \tag{5.22}$$

where the mean and covariance functions are defined as

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}\big\{f(\mathbf{x})\big\} \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}\big\{\big(f(\mathbf{x}) - m(\mathbf{x})\big)\big(f(\mathbf{x}') - m(\mathbf{x}')\big)\big\}. \end{aligned} \tag{5.23}$$

By choosing the mean and covariance functions, any knowledge regarding the function can be incorporated into the prior. Based on this prior and a set of training samples $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$ that are related through f, the objective in this section is to find a description of $f(\mathbf{x}_*)$, where $\mathbf{x}_*$ is an arbitrary vector in $\mathbb{R}^n$.

As often in reality, the observations are affected by noise. This noise is modelled as Gaussian such that each training sample is

$$y^{(i)} = f(\mathbf{x}^{(i)}) + w^{(i)}, \tag{5.24}$$

where $w^{(i)} \sim \mathcal{N}(0, \sigma_n^2)$. From the definition of a Gaussian process, the noisy samples from the process are jointly Gaussian, implying that

$$\mathbf{y} \;\sim\; \mathcal{N}\big(m(\mathbf{x}), \mathbf{K} + \sigma_n^2 \mathbf{I}_{N\times N}\big), \tag{5.25}$$

where $\mathbf{y} = [y^{(1)}, \dots, y^{(N)}]^T$, $m(\mathbf{x}) = [m(\mathbf{x}^{(1)}), \dots, m(\mathbf{x}^{(N)})]^T$ and

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}) & k(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) & \dots & k(\mathbf{x}^{(1)}, \mathbf{x}^{(N)}) \\ k(\mathbf{x}^{(2)}, \mathbf{x}^{(1)}) & k(\mathbf{x}^{(2)}, \mathbf{x}^{(2)}) & \dots & k(\mathbf{x}^{(2)}, \mathbf{x}^{(N)}) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}^{(N)}, \mathbf{x}^{(1)}) & k(\mathbf{x}^{(N)}, \mathbf{x}^{(2)}) & \dots & k(\mathbf{x}^{(N)}, \mathbf{x}^{(N)}) \end{bmatrix}. \tag{5.26}$$

Given the training samples and the prior knowledge regarding the function, the output for an arbitrary input vector can be estimated. Denoting a set of M input vectors by $\mathbf{x}_* = [(\mathbf{x}_*^{(1)})^T, \dots, (\mathbf{x}_*^{(M)})^T]^T$, we wish to find the corresponding outputs $\mathbf{y}_* = [y_*^{(1)}, \dots, y_*^{(M)}]^T$. From the definition of the Gaussian process, the vector of training outputs, $\mathbf{y}$, and the sought outputs $\mathbf{y}_*$ are jointly Gaussian. That is,

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{x}) \\ m(\mathbf{x}_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{bmatrix} \right). \tag{5.27}$$

In [58], it is shown that the conditional distribution $p(\mathbf{y}_*|\mathbf{y})$ is given as

$$p(\mathbf{y}_*|\mathbf{y}) = \mathcal{N}\big(m(\mathbf{x}_*) + \mathbf{K}_*^T \mathbf{K}^{-1}(\mathbf{y} - m(\mathbf{x})), \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_*\big). \tag{5.28}$$

To illustrate the discussed theory, the section ends with an example.

Example 5.2 (Gaussian processes)
In this example we consider an unknown function f, such that $y = f(x)$, where both x and y are scalars. A prior of f can be specified as

$$f \sim \mathcal{GP}(m(x), k(x, x')) \tag{5.29}$$

where $m(x)$ and $k(x, x')$ shall be chosen to summarize our prior knowledge regarding the function. Assume that we know that the function is fairly smooth and varies around $y = 0$. These properties are captured by

$$m(x) \;=\; 0 \tag{5.30}$$

$$k(x, x') \;=\; exp\left(-\frac{1}{2L^2}(x - x')^2\right), \tag{5.31}$$

where $L = 3$. Changing L affects the smoothness of the function.

For illustration, we study the function on a limited interval by constructing a vector $\mathbf{x}$ *that consists of* n *equally spaced values on* $[-30, 30]$*. A sample from the process in* (5.29) *is a function [58]:*

$$f_s(\mathbf{x}) = m(\mathbf{x}) + \sqrt{\mathbf{K}} \cdot \xi, \tag{5.32}$$

where $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n\times n})$*, and* $\mathbf{K}$ *is a matrix whose elements are given by* $\mathbf{K}(i,j) = k(x_i, x_j)$ *as defined in* (5.31)*. In Figure 5.2, two samples from the process are depicted.*

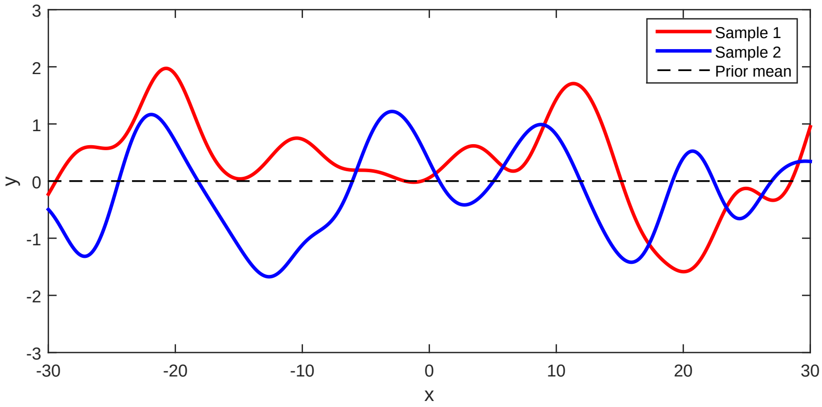


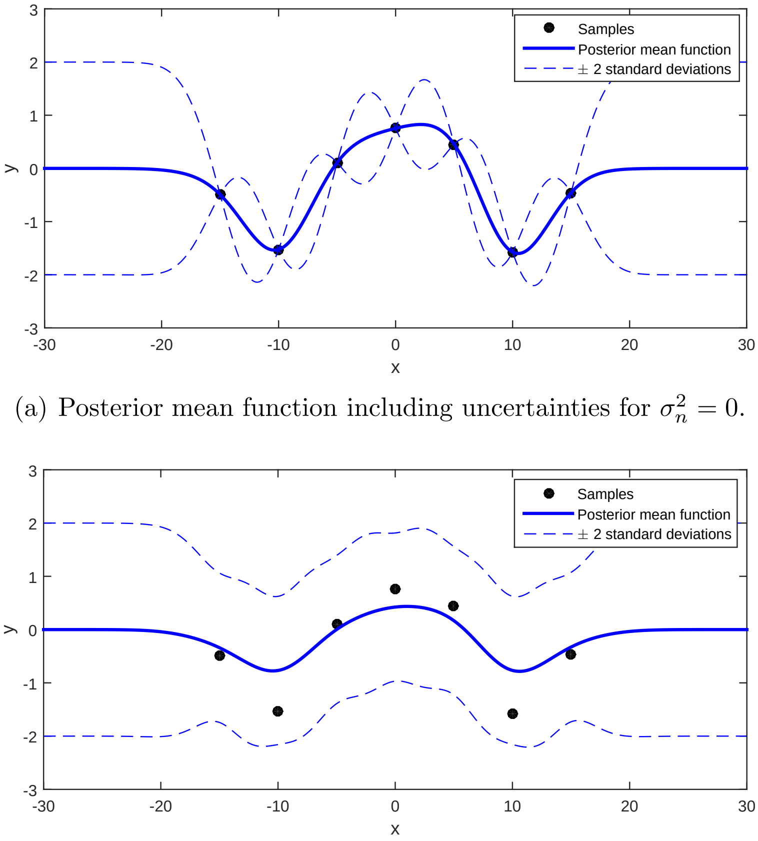
Figure 5.2: Samples from the prior process.

Assume that we observe the function for 7 different values of x*. For* $i = 1, 2, \ldots, 7$*, these samples can be described as*

$$y^{(i)} = f(x^{(i)}) + w^{(i)}, \tag{5.33}$$

where $w_{(i)} \sim \mathcal{N}(0, \sigma_n^2)$ *is the observation noise. Depending on the noise level, the information regarding* f *obtained from the samples will differ. For illustration, we consider two scenarios. First, the noise variance is* $\sigma_n^2 = 0$*, i.e., we observe the true function values, and second,* $\sigma_n^2 = 1$*.*

The estimated functions, including uncertainties, are shown in Figure 5.3. Worth noting in 5.3(a) is that when the samples are noise-free, the true function is observed and consequently there are no uncertainties regarding the function at these points. This can be compared to the case where the samples are noisy. Then, as shown in Figure 5.3(b), the samples provide less information about the underlying function. Finally, in both figures it is clear that in unobserved intervals, such as $x \in [20, 30]$*, the posterior function is very similar to the prior. The reason is that the available samples do not provide much information regarding the function on these intervals.*



(a) Posterior mean function including uncertainties for $\sigma_n^2 = 0$.

(b) Posterior mean function including uncertainties for $\sigma_n^2 = 1$.

Figure 5.3: The posterior functions for training samples with two different noise levels.

Bibliography

[1] F. Gustafsson, “Automotive Safety Systems,” *IEEE Signal Processing Magazine*, vol. 26, pp. 32–47, July 2009.

[2] M. Bertozzi, A. Broggi, and A. Fascioli, “Vision-based intelligent vehicles: State of the art and perspectives,” *Robotics and Autonomous Systems*, vol. 32, no. 1, pp. 1 – 16, 2000.

[3] K. Kozak, J. Pohl, W. Birk, J. Greenberg, B. Artz, M. Blommer, L. Cathey, and R. Curry, “Evaluation of lane departure warnings for drowsy drivers,” in *Proceedings of the human factors and ergonomics society annual meeting*, vol. 50, pp. 2400–2404, Sage Publications, 2006.

[4] E. Coelingh, A. Eidehall, and M. Bengtsson, “Collision Warning with Full Auto Brake and Pedestrian Detection - a practical example of Automatic Emergency Braking,” in *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 155–160, September 2010.

[5] M. Distner, M. Bengtsson, T. Broberg, and L. Jakobsson, “City safety – a system addressing rear-end collisions at low speeds,” in *Proc. 21st International Technical Conference on the Enhanced Safety of Vehicles*, no. 09-0371, 2009.

[6] P. Dollár, C. Wojek, B. Schiele, and P. Perona, “Pedestrian Detection: An Evaluation of the State of the Art,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 743–761, April 2012.

[7] D. Geronimo, A. Lopez, A. Sappa, and T. Graf, “Survey of Pedestrian Detection for Advanced Driver Assistance Systems,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1239–1258, July 2010.

[8] D. Forslund and J. Bjärkefur, “Night Vision Animal Detection,” in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 737–742, June 2014.

[9] “Preliminary Statement of Policy Concerning Automated Vehicles.” http://www.nhtsa.gov, May 2013.

[10] A. Eugensson *et al.*, “Environmental, safety, legal and societal implications of autonomous driving systems,” in *The 23rd Enhancement Safety of Vehicles Conference*, 2013.

[11] C. Urmson *et al.*, “Autonomous driving in urban environments: Boss and the Urban Challenge,” *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[12] S. Kammel *et al.*, “Team AnnieWAY’s autonomous system for the 2007 DARPA Urban Challenge,” *Journal of Field Robotics*, vol. 25, no. 9, pp. 615–639, 2008.

[13] E. Guizzo, “How Google’s self-driving car works,” *IEEE Spectrum Online, October*, vol. 18, 2011.

[14] J. Ziegler *et al.*, “Making Bertha Drive – An Autonomous Journey on a Historic Route,” *IEEE Intelligent Transportation Systems Magazine*, vol. 6, pp. 8–20, Summer 2014.

[15] G. Brooker, M. Bishop, and S. Scheding, “Millimetre waves for robotics,” in *Australian Conference for Robotics and Automation*, 2001.

[16] M. I. Skolnik, *Radar handbook*. New York: McGraw-Hill, 2008.

[17] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt, “Millimeter-wave technology for automotive radar sensors in the 77 GHz frequency band,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 3, pp. 845–860, 2012.

[18] W. Fleming, “New Automotive Sensors – A Review,” *IEEE Sensors Journal*, vol. 8, pp. 1900–1921, November 2008.

[19] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS – global navigation satellite systems: GPS, GLONASS, Galileo, and more.* Springer Science & Business Media, 2007.

[20] R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82 (Series D), pp. 35–45, 1960.

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

[22] B. Bell and F. Cathey, "The iterated Kalman filter update as a Gauss-Newton method," *IEEE Transactions on Automatic Control*, vol. 38, pp. 294–297, February 1993.

[23] G. Sibley, G. Sukhatme, and L. Matthies, "The Iterated Sigma Point Kalman Filter with Applications to Long Range Stereo.," in *Robotics: Science and Systems*, vol. 8, pp. 235–244, 2006.

[24] S. J. Julier and J. K. Uhlmann, "New Extension of the Kalman Filter to Nonlinear Systems," *Proc. SPIE*, vol. 3068, pp. 182–193, 1997.

[25] I. Arasaratnam and S. Haykin, "Cubature Kalman Filters," *IEEE Transactions on Automatic Control*, vol. 54, pp. 1254–1269, June 2009.

[26] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings of Radar and Signal Processing*, vol. 140, pp. 107–113, April 1993.

[27] M. S. Arulampalam, S. Maskell, and N. Gordon, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Ttransactions on Signal Processing*, vol. 50, pp. 174–188, 2002.

[28] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[29] Y. Bar-Shalom and E. Tse, "Tracking in a Cluttered Environment with Probabilistic Data Association," *Automatica*, vol. 11, no. 5, pp. 451 – 460, 1975.

[30] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association," *IEEE Journal of Oceanic Engineering*, vol. 8, pp. 173 – 184, July 1983.

[31] D. B. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[32] R. P. Mahler, *Statistical Multisource-Multitarget Information Fusion.* Artech House, Inc., 2007.

[33] R. Mahler, "Multitarget Bayes Filtering via First-Order Multitarget Moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[34] O. Erdinc, P. Willett, and Y. Bar-Shalom, "Probability Hypothesis Density Filter for Multitarget Multisensor Tracking," in *8th International Conference on Information Fusion*, vol. 1, July 2005.

[35] R. Mahler, "PHD Filters of Higher Order in Target Number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, 2007.

[36] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[37] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Analytical Implementations of the Cardinalized Probability Hypothesis Density Filter," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, 2007.

[38] R. Mahler, "PHD Filters for Nonstandard Targets, I: Extended Targets," in *12th International Conference on Information Fusion, 2009. FUSION '09.*, pp. 915 –921, July 2009.

[39] K. Granström and U. Orguner, "A PHD Filter for Tracking Multiple Extended Targets Using Random Matrices," *IEEE Transactions on Signal Processing*, vol. 60, pp. 5657–5671, November 2012.

[40] C. Lundquist, K. Granström, and U. Orguner, "An Extended Target CPHD Filter and a Gamma Gaussian Inverse Wishart Implementation.," *Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 472–483, 2013.

[41] C. Lundquist, L. Hammarstrand, and F. Gustafsson, "Road Intensity Based Mapping Using Radar Measurements With a Probability Hypothesis Density Filter," *IEEE Transactions on Signal Processing*, vol. 59, pp. 1397–1408, April 2011.

[42] M. Adams, J. Mullane, E. Jose, and B.-N. Vo, *Robotic Navigation and Mapping with Radar.* Artech House, 2012.

[43] J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo, “A Random-Finite-Set Approach to Bayesian SLAM,” *IEEE Transactions on Robotics*, vol. 27, pp. 268–282, April 2011.

[44] R. Mahler, B.-T. Vo, and B.-N. Vo, “CPHD Filtering With Unknown Clutter Rate and Detection Profile,” *IEEE Transactions on Signal Processing*, vol. 59, pp. 3497 –3513, August 2011.

[45] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, “A Consistent Metric for Performance Evaluation of Multi-Object Filters,” *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.

[46] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum Likelihood From Incomplete Data Via the EM Algorithm,” *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[47] K. P. Murphy, *Machine Learning: a Probabilistic Perspective.* MIT press, 2012.

[48] S. Borman, “The Expectation Maximization Algorithm – a Short Tutorial.” http://www.seanborman.com/publications/EM_algorithm.pdf.

[49] C. M. Bishop, *Pattern recognition and machine learning*, vol. 1. Springer, New York, 2006.

[50] M. J. Beal, *Variational Algorithms For Approximate Bayesian Inference.* PhD thesis, University of London, 2003.

[51] M. J. Beal and Z. Ghahramani, “The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures,” *Bayesian statistics*, vol. 7, pp. 453–464, 2003.

[52] S. Kullback and R. A. Leibler, “On Information and Sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[53] T. Minka *et al.*, “Divergence measures and message passing,” tech. rep., Microsoft Research, 2005.

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

[55] C. Wang and D. M. Blei, “Variational Inference in Nonconjugate Models,” *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1005–1031, 2013.

[56] D. M. Blei, M. I. Jordan, *et al.*, “Variational inference for Dirichlet process mixtures,” *Bayesian analysis*, vol. 1, no. 1, pp. 121–143, 2006.

[57] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning.* The MIT Press, 2006.

[58] C. E. Rasmussen, “Gaussian processes in machine learning,” in *Advanced lectures on machine learning*, pp. 63–71, Springer, 2004.