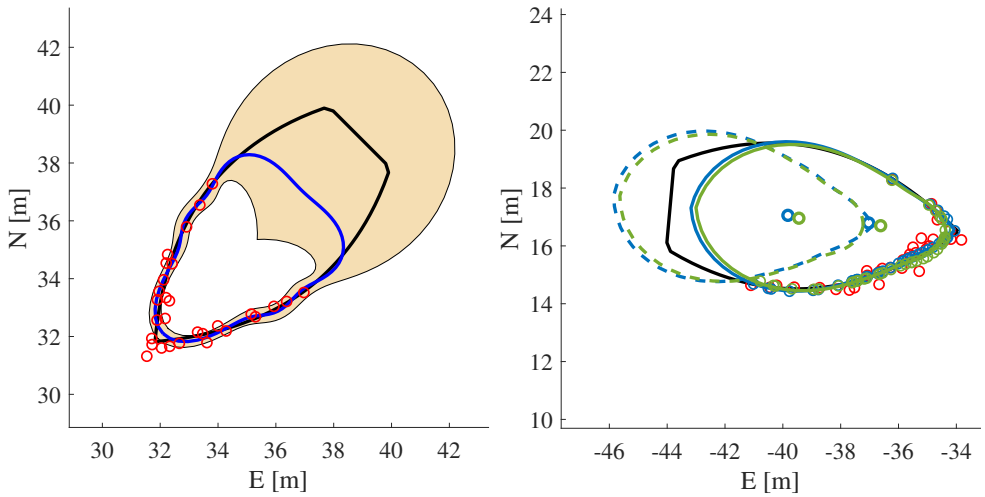
NTNU – Trondheim
Norwegian University of
Science and Technology

TTK4550 - Specialization Project

LIDAR Extended Object Tracking of a Maritime Vessel using Gaussian Processes

Author: Michael Ernesto López
Supervisor: Edmund Førland Brekke
Submission date: June 2019



Abstract

An important part of autonomous ship systems, such as collision avoidance systems, is to determine the movement, size and shape of other objects in the vicinity.

Extended object tracking (EOT) is the problem of estimating the movement and the extent of an object based on high-resolution measurements, such as lidar measurements, while Gaussian processes (GPes) offer a stochastic model for parameterizing the object's extent, and a well-established regression technique.

In this study, a single object that resembles the hull of a maritime vessel is considered. Different GP models for the object's extent are studied, and Kalman filters are developed using these GP models together with lidar measurements to solve the EOT problem for this object.

Among others, an original GP that models axis symmetric extents, as well as an original extended Kalman filter based on implicit derivation are presented.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AIS	=	Automatic Identification System
ANEES	=	Average Normalized Estimation Error Squared
ASV	=	Autonomous Surface Vehicle
EKF	=	Extended Kalman Filter
EOT	=	Extended Object Tracking
GNSS	=	Global Navigation Satellite System
GP	=	Gaussian Process
GPDA	=	General Probabilistic Data Association
KF	=	Kalman Filter
LIDAR	=	Light Detection and Ranging
NED	=	North-East-Down (frame)
PHD	=	Probability Hypothesis Densisty
RADAR	=	Radio Detection and Ranging
RMSE	=	Root Mean Square Error

Chapter 1

Introduction

1.1 Background and motivation

The ability to estimate the position and direction of vessels, as well as to detect other objects in the vicinity and predict their movement, has been crucial for successful seafaring since the dawn of time. Throughout history, different instruments and techniques have been developed to help the crew navigate by enhancing their situational awareness, such as the classical celestial navigation, charts, sextants, compasses and light signals to a plethora of modern sensors and systems such as gyroscopes, radars, sonars, global navigation satellite systems (GNSS), automatic identification systems (AIS), etc. Nevertheless, lookouts have always been heavily relied upon in maritime navigation, and humans are still mainly responsible for integrating the available navigational information in order to guide their vessels. However, this is starting to change.

In the last years, the continuous growth in processing power and the access to more capable and affordable hardware has fueled the interest of several industries in automating ever more complex processes with the ultimate goal of minimizing -if not eliminating- the necessity of human intervention for their proper operation. This is also a trend present in maritime operations, where autonomous surface vehicles (ASV) are of major interest. In Norway alone, there are several ASV projects under development, such as the container ship Yara Birkeland from Kongsberg and Yara, which is scheduled to be operational next year [7, 8], the container ship ReVolt from DNV-GL [5] and the Telemetron developed by Maritime Robotics. Autonomous maritime vessels constitute also a field of interested for academic institutions. Here at NTNU, there are the research projects Autoferry [1], which develops an autonomous electric ferry for pedestrian and cyclist transport across a canal in Trondheim, and the Autosea project [2], which conducts research into sensor fusion and collision avoidance methods for autonomous ships.

All autonomous ship projects have common major challenges that arise from the elimination of the human factor in navigation and guidance. One of these challenges is to use sensor data to first detect objects in the vicinity of the vessel, and then proceed to track them by continuously estimating their kinematic quantities, such as position, velocity, heading and turn rate. This is known as object tracking or target tracking. If the resolution of the sensors on-board is high enough, several measurements from a tracked object could be available at a given time. Such situation is referred to as Extended Object Tracking (EOT), and the multiple measurements can be used to estimate the geometry of the object, also known as the object's extent.

Knowledge about the extent should play an important role in identification and collision avoidance systems. Since even in the particular case of the object being a maritime vessel, there is a plethora of possible shapes and sizes for its hull, a versatile EOT method that can handle this diversity will be very appealing for autonomous maritime operations. Gaussian Processes (GPes) have the potential to provide such a method, and their integration into EOT is the main focus of this report.

Gaussian Processes originated as a Bayesian regression technique for estimating the values of an unknown function given some isolated measurements. In our context, the unknown function represents the contour of the object being tracked. In contrast to many other methods, Gaussian Processes provide a non-parametric model in the sense that no assumptions are made on the underlying structure of the unknown function. Furthermore, this model is stochastic, and can therefore be smoothly merged with probabilistic filters, which are extensively used in EOT. This enables integration of prior and uncertainty information.

Finally, we would like to mention that EOT can be performed with a variety of high-resolution sensors, such as radars, lidar sensors, cameras and infrared sensors, which can be combined in different sensor configurations. Ideally, several sensor of different types should be fused in order to benefit from the strengths of each sensor and mitigate their individual shortcomings. However, sensor fusion methods (see for example [31]) are out of the scope of this work, and only lidar measurements will be used in the simulations presented in this report.

1.2 Literature review

In this section, the relevant articles/papers and books for this report are presented and discussed. We start by reviewing literature about the topics of Gaussian Processes and EOT. Finally, we discuss some publications that combine both fields.

The theoretical background for Gaussian Processes and their regression technique were developed by N. Wiener [29] and A.N. Kolmogorov [20] in the 1940s. However, the alleged first major application can be traced back to the work of Krige [21] in 1951, who developed an interpolating technique of geostatistical data since the methods used at the time gave unsatisfactory results. Therefore Gaussian Process regression is also known as *Wiener-Kolmogorov prediction* or *Kriging*.

While Gaussian Processes are appealing for their non-parametric modeling of functions, their main drawback is the high computational cost of regression, which usually involves the inversion of a large matrix. However, due to the continuous growth in processing power, this drawback is no longer a major restriction, and applications of Gaussian Processes have moved from spatial statistics to many other fields, where optimization and prediction is crucial. One of these fields is *Machine Learning*, which is the main focus application in the standard textbook "Gaussian Processes for Machine Learning" [23] by Rasmussen and Williams. This book has served as the foundation of Chapter 4 of this report.

The first contribution in EOT can be traced back to 1988 in the work of Drummond et al.

in [11]. Many other contributions to the field have been added since. In the recent tutorial paper [16], Granstrom et al. formulates the EOT problem in a multiple target setting, and give an overview of the current state-of-the-art research in the field up to the article's publication in 2017. Here, the two most used EOT approaches are discussed: The random matrix approach and the Kalman filter approach for star-convex shapes, which are both Bayesian approaches.

In the random matrix approach, the object's state consists of kinematic variables that describe the movement of the object, such as position, velocity, heading or turn rate, and of a matrix that describes the object's extent. This matrix is modeled as a symmetric and positive definite matrix, which is 2-by-2 or 3-by-3 depending on the spatial dimensions of the problem. In other words, the extent is approximated by an ellipse or an ellipsoid, respectively. The random matrix approach was first proposed by Koch in [19], where a Bayesian approach was used to estimate the elliptical extent. In this work, there was a restrictive coupling between the state that describe the kinematics and the matrix that describes the extent, and the sensor noise was not part to the model. These issues were later addressed by Feldmann et al. in [12]. In [14], Granström and Orguner made another improvement by modeling the possible rotations of elliptical extent. In [26], Schuster and Reuter used the general probabilistic data association (GPDA) filter, which was developed by Schuster et al. in [27], to handle clutter measurements. In this work, experiments results with lidar and radar measurements were compared. In [15], Granström et al. used another probabilistic filter to handle clutter measurements: the probability hypothesis density (PHD) filter. The developed method was used to estimate the contour of multiple rectangular and elliptical objects using lidar measurements. Inspired by this, Ruud combined the sensor model from [15] with the GPDA approach from [27] in his MSc thesis [24] and related conference paper [25]. The developed method was applied on real lidar data from a passenger ferry in Trondheim, whose hull shape can be fairly approximated by an ellipse.

The random matrix approach may work very well for tracking objects, whose extent can be approximated by ellipses or rectangles, such as cars or pedestrians. However, in applications were this is not a reasonable assumption or a better estimate of the extent is desirable, star-convex shapes may be considered as an alternative parametrization. In this approach, the object's state consists of variables that describe the kinematics of the object, as in the random matrix approach, together with multiple variables that describe the object's extent. The latter variables are related to the existence of a radius function that describes the boundary of the extent. The existence of this function is guaranteed if the extent is star-convex and compact. Note that the compactness assumption is not mentioned in the literature since the extent is tacitly assumed to be closed and bounded.

The two main frameworks for representing the star-convex extent are based on Random Hypersurface models and Gaussian Process models. If the Random Hypersurface model is used, the boundary of the extent is assumed to be given by an implicit equation, and the variables that describe the extent are the Fourier coefficients of the radius function that parameterizes the extent's boundary. The more Fourier coefficients are added to the object's state, the more detailed the estimate of the extent can be. In [9], Baum and Hanebeck present an EOT method using Random Hypersurface models that is suited for radar measurements. On the other hand, if the Gaussian Process model is used, then the states that describe the extent are just values of the radius function. Again, the finer this discretization of the boundary is, the more detailed the estimated extent can be. In [28], Wahlström and Özkan used this approach to develop an EOT

method for lidar measurements and radar measurements. The study of this method and critical investigation of its performance for maritime vessels is the main objective of this report.

1.3 Problem description and assumptions

The movement and shape of a single two-dimensional object is to be estimated using the measurements provided by a single lidar sensor. The object's extent is a star-convex and compact rigid body, and resembles the hull of a maritime vessel. Both the movement of the object and the lidar measurements are simulated. Testing all real data falls outside the scope. In the simulations, the position of the lidar sensor does not vary with time. Furthermore, all generated lidar measurements correspond to a point in the extent. In other words, data association uncertainty related to misdetections or measurement clutters is not considered in this problem.

In this report, we study the EOT method proposed in [28] to solve this problem. All relevant aspects of this method are introduced and discussed. Furthermore, some suggestions to this method are proposed in order to contribute to the discussion and to study the potential of Gaussian Processes for EOT using lidar measurements.

1.4 Outline

The report is organized as follows:

- In Chapter 2, we introduce the world frame and the local body frame for the object, and define the state of the object. Finally, some models for the extent that resemble a hull are presented. The mathematical details are added for completion, and may be skipped without loss of continuity.
- In Chapter 3, we introduce lidar sensors, and discuss how lidar measurements are simulated with special emphasis in the distribution of the measurement error.
- In Chapter 4, Gaussian Processes and their regression technique are introduced. Special attention is given to Gaussian Processes that model closed curves, closed curves that are symmetric about a point and closed curves that are symmetric about an axis.
- In Chapter 5, state-space formulations for the EOT problem at hand are derived. Two situations are discussed. In the first situation, the kinematic variables of the object's state are known, and a linear Kalman Filter is used for inference. In the second situation, the entire state is unknown, and an extended Kalman Filter is used for inference.
- In Chapter 6, the inference methods developed in the previous chapter are tested in different simulations. The obtained results are discussed.
- In Chapter 7, we present the conclusions of this report and recommendations for further work.
- In Appendix A, a result about multivariate normal distributions used in Chapter 4 is presented.

- In Appendix B, we present the discretization of a continuous velocity model, the algorithms for the linear and extended Kalman filters, as well as tools for studying filter consistency.
- In Appendix C, D and E, all the figures with the obtained results for the different simulations from Chapter 6 are presented.

Chapter 2

Reference frames and the object state

2.1 The world and body frames

All the simulations take place in a fixed world frame, which is denoted by $\{w\}$. Since these simulations occur over a relatively short time-span, and the tracked object does not move relatively far away, nor fast, the world frame can be assumed to be a local North-East-Down (NED) frame, which is in addition inertial. NED frames are standard local reference frames, which are for example introduced in [13, ch. 2]. Intuitively, they can be located at any point of the Earth's surface, and their 2 first axes correspond to the North and East directions in a projected chart. The third axis completes the right-handed system, and therefore points "down". A local NED frame is illustrated in Figure 2.1.

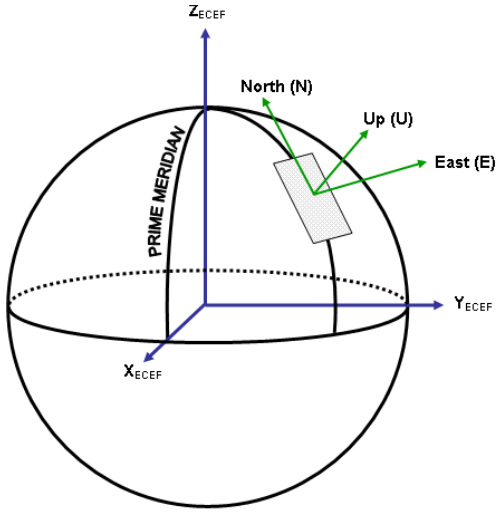


Figure 2.1: Local North-East-Down frame (green) at a point on the Earth's surface. Source: [3].

Since the EOT problem is restricted to the two dimensions of the horizontal surface, only the North and East coordinates are relevant. Hence, the position of a point in the world frame is denoted by

$$\mathbf{r}^w = \begin{bmatrix} N & E \end{bmatrix}^T , \tag{2.1}$$

where N and E are the displacement in meters along the the North and East directions respect to the origin of the world frame. In other words, N and E correspond to the difference in longitude and latitude between the point and the origin, respectively.

When studying the movement of a rigid body such as the tracked object, it is advantageous to define a local frame that is attached to this body and follows its movement. This is known as the *body frame*, and is denoted by $\{b\}$. Usually, the origin of the body frame is a fixed point in the rigid body, and the frame axes correspond to existing body symmetries, if any. In the case of a maritime vessel such as the tracked object, the first body axis is chosen to give the stern to bow direction.

Since the body frame is attached to the rigid body, the position and orientation of the rigid body respect to the fixed world frame is completely determined by the body frame. The position is given by the origin of the body frame, while the orientation is given by the axes of the body frame. Furthermore, the orientation of the rigid body can be described by a set of three angles, known as *Euler angles* (see [13, ch. 2]). However, in the two-dimensional case, only one angle is necessary, and in the case of maritime vessels, the *heading angle* is chosen. The heading angle ψ_c is the angle formed between the first axis of the world frame (North direction) and the first axis of the body frame (stern to bow direction). In other words, the heading angle corresponds to the direction the bow points in, hence the name.

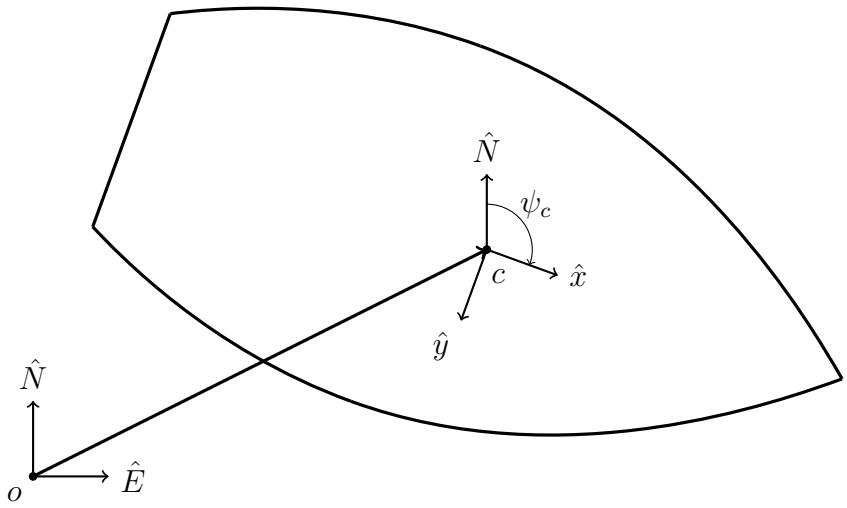


Figure 2.2: A maritime vessel with heading ψ_c. The world frame with origin o and axes $\hat{N}$ (North) and $\hat{E}$ (East), and the body frame with origin c and axes $\hat{x}$ and $\hat{y}$.

The relation between the world and body frames is illustrated in Figure 2.2. As one can see, the transformation from the world to the body frame is the composition of a rotation of ψ_c about the "down" axis, followed by a translation from the origin of the world frame, o, to the origin of the body frame, c. In symbols,

$$\mathbf{r}^w = \mathbf{r}_c^w + \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \mathbf{r}^b \,, \tag{2.2}$$

where $\mathbf{r}_c^w = [N_c, E_c]^T$ are the coordinates of c in the world frame and $\mathbf{r}^w$ and $\mathbf{r}^b$ are the coordinates of an arbitrary point in the world and body frames, respectively.

Therefore, the variables N_c, E_c and ψ_c describe the position and orientation of the object. Furthermore, their derivatives

$$v_E = \dot{E} \tag{2.3a}$$

$$v_N = \dot{N} \tag{2.3b}$$

$$r_c = \dot{\psi}_c \tag{2.3c}$$

describe the *linear velocity*, $[v_N, v_E]^T$, and the *angular velocity*, r_c, of the object.

By putting all these variables together, we have a state vector that can be used to model the motion of the object:

$$\mathbf{x}_c = \begin{bmatrix} N_c & v_N & E_c & v_E & \psi_c & r_c \end{bmatrix}^T. \tag{2.4}$$

Note that this choice of state vector $\mathbf{x}_c$ allows the direction of the linear velocity $[v_N, v_E]^T$ and the heading ψ_c to be different. In other words, $\mathbf{x}_c$ may be used to model sideslip.

2.2 Star-convex extent

As stated in section 1.3, the object's extent, which is denoted by E, is assumed to be a compact (bounded and closed) star-convex subset of $\mathbb{R}^2$. Therefore, the extent is completely determined by its boundary, which is a closed curve.

The star-convexity of E means that there exists a point $\mathbf{r}_0^w \in E$ such that the line segment from this point to any other point in E is always contained in E. In symbols,

$$[\mathbf{x}_0^w, \mathbf{x}^w] = \{\lambda \mathbf{x}_0^w + (1-\lambda)\mathbf{x}^w : \lambda \in [0,1]\} \subset E \tag{2.5}$$

for all $\mathbf{x}^w \in E$. The point given by $\mathbf{r}_0^w$ is called the *center*.

By taking the center as the origin of the body frame, i.e. $\mathbf{r}_c^w = \mathbf{r}_0^w$, the extent's boundary can be described by the following polar parametrization in the body frame:

$$r(\theta) = \max\left\{r \geq 0 : \mathbf{r}^b = [r\cos(\theta), r\sin(\theta)]^T \text{ and } \mathbf{r}^w \in E\right\} \tag{2.6a}$$

$$= \max\left\{r \geq 0 : [N_c + r\cos(\theta+\psi_c), E_c + r\sin(\theta+\psi_c)]^T \in E\right\}. \tag{2.6b}$$

The value $r(\theta)$ is well defined for all θ since E is compact, and the equivalence between eq. (2.6a) and eq. (2.6b) follows from eq. (2.2). The polar parametrization $r(\theta)$ is called the *radius function*, and is illustrated in Figure 2.3. Note also that the election of the center as the origin of the body frame motivates the use of the subscript c for this origin.

In practice, it is unnecessary -if not impossible- to obtain an analytical expression for $r(\theta)$ since a discretization of this function is sufficient to describe the extent. Hence, let $\boldsymbol{\theta}_t = [\theta_{t,n}]_{n=1}^{N_t}$ be a fine enough fixed discretization of the interval $[0, 2\pi)$, i.e.

$$0 = \theta_{t,1} < \theta_{t,2} < \cdots < \theta_{t,N_t} < 2\pi$$

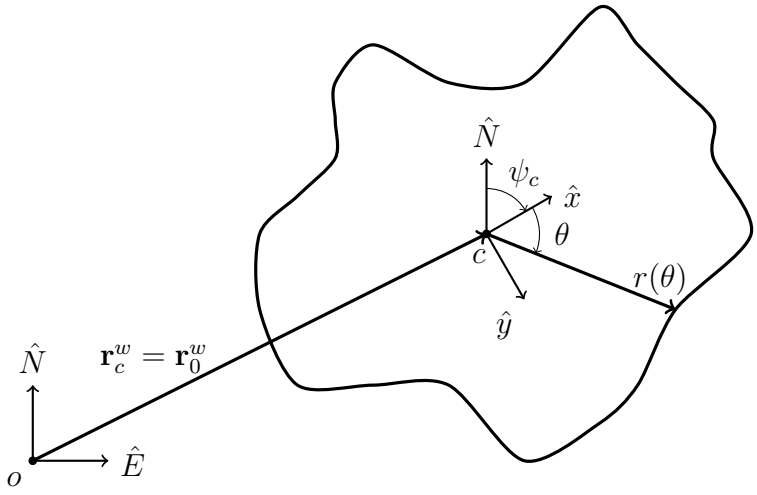


Figure 2.3: Polar parametrization in the body frame of the extent's boundary.

and the maximum step-length

$$\Delta\theta_{t,\max} = \max\left\{\theta_{t,2} - \theta_{t,1}, \cdots, \theta_{t,N_t} - \theta_{t,N_t-1}, 2\pi - \theta_{t,N_t}\right\}$$

is small enough. Then the extent is described by the values of the radius function $r(\cdot)$ at these points, i.e. by the vector

$$\mathbf{x}_t = \begin{bmatrix} r(\theta_{t,1}) & r(\theta_{t,2}) & \cdots & r(\theta_{t,N_t}) \end{bmatrix}^T. \tag{2.7}$$

The angles in $\boldsymbol{\theta}_t = [\theta_{t,n}]_{n=1}^{N_t}$ are called *test angles*, and their name is motivated by the applications of Gaussian Processes to Machine Learning (see [23] or Chapter 4 of this report).

The augmentation of $\mathbf{x}_c$ (eq. (2.4)) with $\mathbf{x}_t$ (eq. (2.7)) gives the *state of the object*

$$\mathbf{x} = \begin{bmatrix} x_c & u_c & y_c & v_c & \psi_c & r_c & r(\theta_{t,1}) & r(\theta_{t,2}) & \cdots & r(\theta_{t,N_t}) \end{bmatrix}^T, \tag{2.8}$$

which describes both the kinematics of the object and the geometry of its extent.

2.3 Hull shape models

The objects that we intend to track, are assumed to resemble the hull of a small ship. This hull is symmetric about its longitudinal axis, and it has an overall length L and a beam B. The width of the beam is achieved at a distance D from the bow. Furthermore, the hull ends in a flat stern of width S. All these specifications are illustrated in Figure 2.4.

Three different methods are considered to find the polar parametrization of the hull with the center placed at midships, i.e. on the longitudinal axis at a distance of $\frac{L}{2}$ from the bow. One method uses spline interpolation for the sides of the hull (starboard and port), the other two consider the sides as piece-wise parabolas or as piece-wise ellipses. We refer to these methods as the "spline" method, the "parabola" method and the "ellipse" method, respectively.

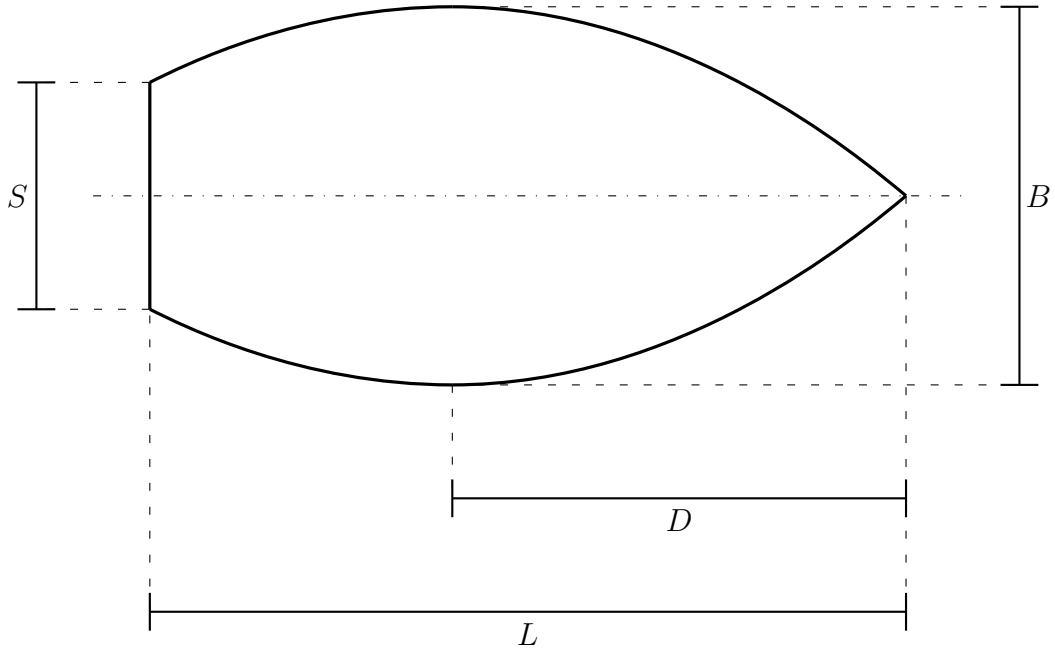


Figure 2.4: Specifications for the hull of the object: Symmetry and parameters L, B, D and S.

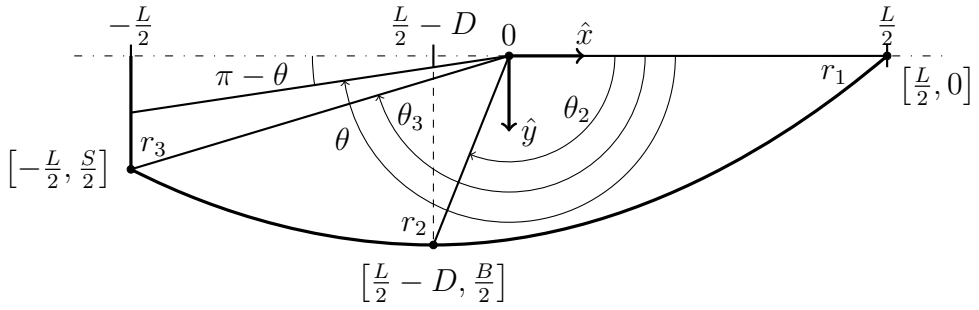


Figure 2.5: Radius function for the hull of the object: Interpolation parameters.

For all these methods the points with coordinates in the body frame $[0, \frac{L}{2}]$, $[\frac{L}{2} - D, \frac{B}{2}]$ and $[-\frac{L}{2}, \frac{S}{2}]$, and their respective polar coordinates:

$$[\theta_1, r_1] = \left[0, \frac{L}{2}\right] \tag{2.9a}$$

$$[\theta_2, r_2] = \left[\operatorname{atan2}\left(\frac{B}{2}, \frac{L}{2} - D\right), \sqrt{\left(\frac{B}{2}\right)^2 + \left(\frac{L}{2} - D\right)^2}\right] \tag{2.9b}$$

$$[\theta_3, r_3] = \left[\operatorname{atan2}\left(\frac{S}{2}, -\frac{L}{2}\right), \sqrt{\left(\frac{B}{2}\right)^2 + \left(\frac{L}{2}\right)^2}\right] ; \tag{2.9c}$$

are of interest for determining the radius function, and are shown in Figure 2.5.

First, the values of the radius function on the interval $[0, \theta_3]$ are determined. For the "spline" method, the knots $[\theta_1, r_1]$, $[\theta_2, r_2]$ and $[\theta_3, r_3]$ are used directly for spline interpolation, which gives the radius function on $[0, \theta_3]$. For the two other methods, cartesian curves for the starboard

side are first derived before finding the radius function.

The cartesian curve for the "parabola" method is

$$y_p(x) = \begin{cases} -\frac{B-S}{2(L-D)^2}\left(x+\frac{L}{2}\right)^2 + \frac{B-S}{L-D}\left(x+\frac{L}{2}\right) + \frac{S}{2} & , x \in \left[-\frac{L}{2}, \frac{L}{2}-D\right) \\ -\frac{B}{2D^2}\left(x-\frac{L}{2}\right)^2 - \frac{B}{D}\left(x-\frac{L}{2}\right) & , x \in \left[\frac{L}{2}-D, \frac{L}{2}\right] , \end{cases} \tag{2.10a}$$

and the curve for the "ellipse" method is

$$y_e(x) = \begin{cases} \sqrt{-\frac{B^2-S^2}{4(L-D)^2}\left(x+\frac{L}{2}\right)^2 + \frac{B^2-S^2}{2(L-D)}\left(x+\frac{L}{2}\right) + \frac{S^2}{4}} & , x \in \left[-\frac{L}{2}, \frac{L}{2}-D\right) \\ \sqrt{-\frac{B^2}{4D^2}\left(x-\frac{L}{2}\right)^2 - \frac{B^2}{2D}\left(x-\frac{L}{2}\right)} & , x \in \left[\frac{L}{2}-D, \frac{L}{2}\right] , \end{cases} \tag{2.11a}$$

The cartesian curves y_p and y_e are continuous in $\left[-\frac{L}{2}, \frac{L}{2}\right]$, differentiable in $\left(-\frac{L}{2}, \frac{L}{2}\right)$ and satisfy

$$0 = y_p\left(\frac{L}{2}\right) = y_e\left(\frac{L}{2}\right) \tag{2.12a}$$

$$\frac{S}{2} = y_p\left(-\frac{L}{2}\right) = y_e\left(-\frac{L}{2}\right) \tag{2.12b}$$

$$\frac{B}{2} = y_p\left(\frac{L}{2}-D\right) = y_e\left(\frac{L}{2}-D\right) \tag{2.12c}$$

$$0 = y_p'\left(\frac{L}{2}-D\right) = y_e'\left(\frac{L}{2}-D\right). \tag{2.12d}$$

Therefore, these curves model the starboard side as specified, and the radius functions for the "parabola" and "ellipse" methods on $[0, \theta_3]$ are found by taking the polar coordinates of $[x, y_p(x)]$ and $[x, y_e(x)]$ for $x \in \left[-\frac{L}{2}, \frac{L}{2}\right]$, respectively.

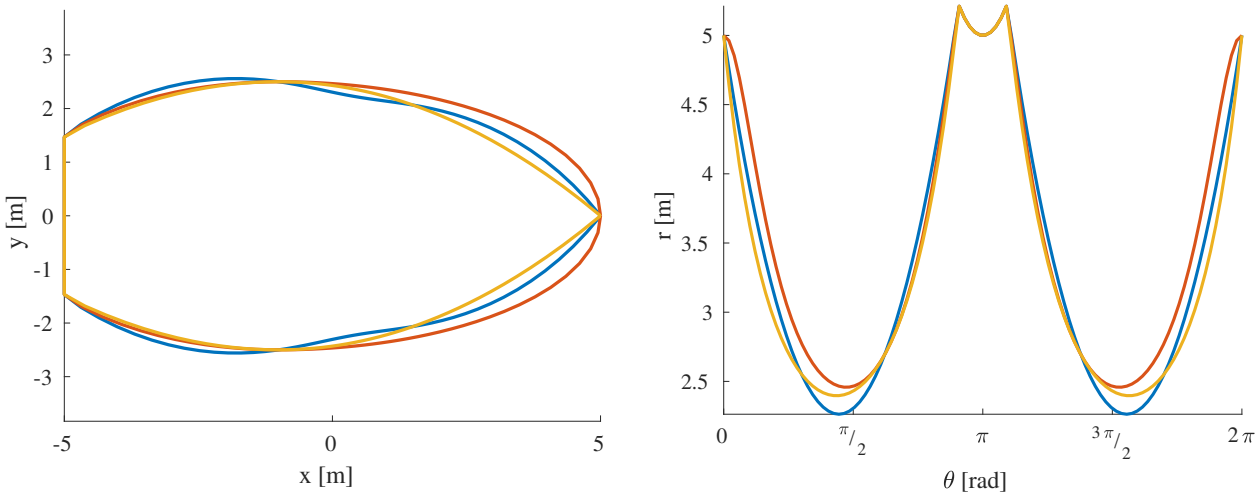
Next, the values of the radius function on the interval $[\theta_3, \pi]$ are determined. This corresponds to the starboard side of the stern. Since the vessel ends in a flat stern as illustrated in Figure 2.5, this gives the trigonometric identity

$$\frac{r(\theta)}{\frac{L}{2}} = \cos(\pi - \theta) = \cos(\theta). \tag{2.13}$$

Hence, the radius function for all the methods is defined on this interval as $r(\theta) = \frac{L}{2}\sec(\theta)$. This completes the parametrization on $[0, \pi]$. Finally, for $\theta \in [\pi, 2\pi]$, the radius function is defined as $r(\theta) = r(2\pi - \theta)$ in order to ensure longitudinal symmetry.

Figure 2.6 shows the different hull shapes and radius functions obtained from each method. All methods used the same values for the hull parameters L, B, D and S, which are summarized in Table 2.1. These values are chosen so that the hull resembles a small ship.

As one can see in Figure 2.6a, the "spline" method has a tendency to give a hull that is bent inward at some points, which does not give a realistic shape. Moreover, the cartesian curve obtained from this method does not necessarily satisfy the specifications. Therefore, the "spline" method is not preferred. On the other hand, the "parabola" and "ellipse" give a realistic hull shape, which resembles a skiff. The hulls are mainly smooth, with some sharp edges. For the "ellipse" method the sharp edges are located at the stern, while the "parabola" method gives an additional sharp edge at the bow. Furthermore, note that the "parabola" and "ellipse" method guarantee a convex extent.



(a) Cartesian curves in the body frame.

(b) Radial functions.

Figure 2.6: Hull shapes obtained from the ”spline” method (blue), the ”parabola” method (yellow) and the ”ellipse” method (red) for the hull parameter values in Table 2.1

Parameter	Value [m]
L	10
B	5
D	6
S	3

Table 2.1: Hull parameter values.

Chapter 3

Lidar measurements

3.1 Lidar sensors

A lidar (light detection and ranging) sensor is a high-resolution active electro-optical sensor that measures the distance to the nearest objects in its surroundings as a function of direction. This is achieved by sending pulsed laser light in several directions covering a region of space. These laser beams propagate through the atmosphere until they hit an object within the sensor's range. The reflected beams are received back at the sensor, and are used to estimate the distance between the sensor and the hit object.

The method for estimating the distance, also known as *range*, depends on the particular type of lidar sensor. Measuring principles may involve the travel time of the light from and back to the sensor, or the difference of physical properties between the emitted and received light waves, such as amplitude or phase changes [18, ch. 9].

(a) Velodyne's Puck™(VLP-16).

(b) Neptec's Opal™3D Lidar.

Figure 3.1: Examples of lidar sensors.

Lidar sensors may also differ in the way they scan their surroundings. For example, the Velodyne's Puck™(VLP-16) [6], which is used as a reference in this report, rotates an array of lasers that scan the entire vertical section of the covered region at the same time. Other lidars may consist of a rotating prism or mirror that reflects a single laser beam in different directions in order to scan the surroundings, such as Neptec's Opal™3D Lidar [4]. Pictures of both examples of lidar sensors are shown in Figure 3.1.

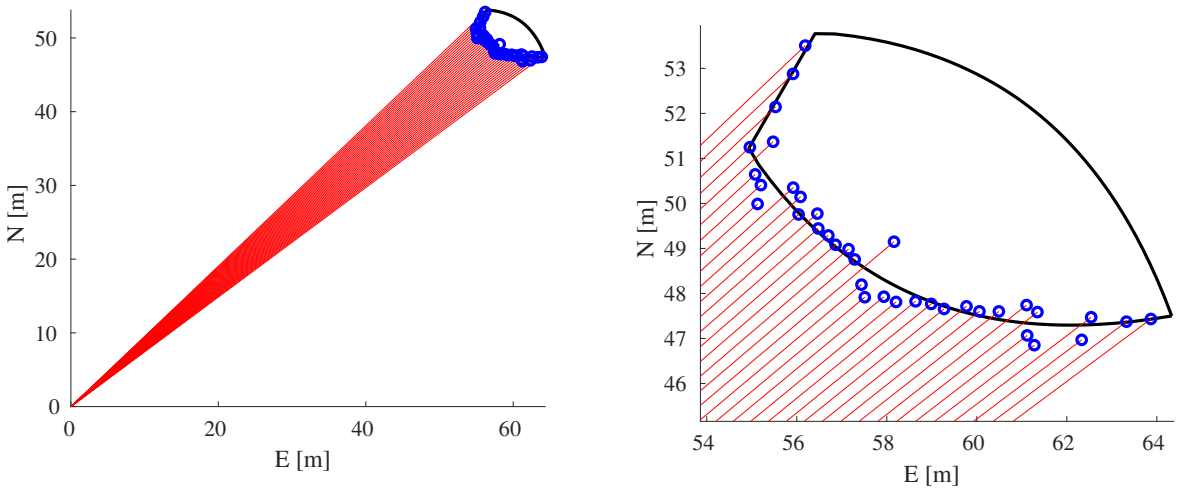
3.2 Lidar measurements used in the simulations

In general, the lidar range measurement are given as a function of direction, which is parameterized using azimuth and elevation angles. This results in a three dimensional point cloud of measurements. However, in our case, only the horizontal dimensions are of interest (see Section 1.3). Hence, the elevation angle is irrelevant, and the range measurements are only a function of the azimuth angle, which is the angle between the laser light beam and the North direction.

Let r_m denote the range measurement for a azimuth angle φ_m. It is assumed that the azimuth angles are known exactly, and that the range measurements can be modelled as

$$r_m = r_{m,\text{true}} + w_r \tag{3.1a}$$
$$w_r \sim \mathcal{N}(0, \sigma_r^2)\,, \tag{3.1b}$$

where $r_{m,\text{true}}$ is the true range value, w_r is the range measurement noise and $\sigma_r > 0$ is the range measurement noise strength. In other words, the range measurements are affected by unbiased normally distributed noise, whose strength is the same for all azimuth angles.



(a) Overview.

(b) Close-up.

Figure 3.2: Example of lidar measurements: laser beams (red), object boundary (black) and points that correspond to lidar range measurements (blue).

Let $\mathbf{r}^w_{\text{lidar}}$ the position of the lidar sensor in the world frame. Then the range measurement r_m with azimuth angle φ_m corresponds to the point

$$\mathbf{z}^w_m = \mathbf{r}^w_{\text{lidar}} + r_m \begin{bmatrix} \cos(\varphi_m) \\ \sin(\varphi_m) \end{bmatrix}\,, \tag{3.2}$$

which is a measurement of the point

$$\mathbf{z}^w_{m,\text{true}} = \mathbf{r}^w_{\text{lidar}} + r_{m,\text{true}} \begin{bmatrix} \cos(\varphi_m) \\ \sin(\varphi_m) \end{bmatrix}\,. \tag{3.3}$$

Therefore, the measurement error of the world frame coordinates is given by

$$\mathbf{z}_m^w - \mathbf{z}_{m,\text{true}}^w = w_r \begin{bmatrix} \cos(\varphi_m) \\ \sin(\varphi_m) \end{bmatrix} , \tag{3.4}$$

which is distributed as

$$\mathbb{E}\left[\mathbf{z}_m^w - \mathbf{z}_{m,\text{true}}^w\right] = \mathbf{0} \tag{3.5a}$$

$$\mathbb{V}\left[\mathbf{z}_m^w - \mathbf{z}_{m,\text{true}}^w\right] = \sigma_r^2 \begin{bmatrix} \cos^2(\varphi_m) & \cos(\varphi_m)\sin(\varphi_m) \\ \cos(\varphi_m)\sin(\varphi_m) & \sin^2(\varphi_m) \end{bmatrix} \tag{3.5b}$$

As mentioned in Section 1.3, only one lidar sensor is used in the simulations presented in this report and it does not move. Hence, it can be placed at the origin of the world frame without loss of generality, i.e. $\mathbf{r}_{\text{lidar}}^w = [0, 0]^T$. Moreover, the lidar measurements are simulated, and the parameters used for these simulations are summarized in Table 3.1, which are inspired by the technical specifications of the Velodyne's Puck (VLP-16) [6]. In the simulations, for each lidar laser beam that intersects the object, the first intersection point is found, and measurement noise is added according to Equation (3.5). Figure 3.2 shows an example of simulated lidar measurements.

Parameter	Value
Measurement range	$100\,\text{m}$
Horizontal angular resolution	$0.2°$
Range noise strength	$0.1\,\text{m}$

Table 3.1: Lidar sensor values used in the simulations.

Chapter 4

Gaussian Processes Fundamentals

4.1 Definitions and notations

A collection of stochastic variables $\{X_i\}_{i\in\mathcal{I}}$ is said to be a *Gaussian process (GP)* with *mean function* $m : \mathcal{I} \to \mathbb{R}$ and *covariance function* $k : \mathcal{I} \times \mathcal{I} \to \mathbb{R}$ if any finite subset of different stochastic variables $\{X_{i_1}, X_{i_2} \cdots, X_{i_N}\}$ has a joint multivariate Gaussian distribution given by

$$\begin{bmatrix} X_{i_1} \\ X_{i_2} \\ \vdots \\ X_{i_N} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(i_1) \\ m(i_2) \\ \vdots \\ m(i_N) \end{bmatrix}, \begin{bmatrix} k(i_1,i_1) & k(i_1,i_2) & \cdots & k(i_1,i_N) \\ k(i_2,i_2) & k(i_2,i_2) & \cdots & k(i_2,i_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(i_N,i_1) & k(i_N,i_2) & \cdots & k(i_N,i_N) \end{bmatrix} \right). \tag{4.1}$$

In such cases, we write $\{X_i\}_{i\in\mathcal{I}} \sim \mathcal{GP}(m, k)$.

In other words, a GP is a generalization of a multivariate Gaussian distribution to an arbitrary number of stochastic variables, where any finite number of these stochastic variables has a joint Gaussian distribution that is determined by an overall mean and covariance function.

To work with GPes, we introduce some convenient short-hand notations:

Let $\{X_i\}_{i\in\mathcal{I}} \sim \mathcal{GP}(m, k)$ be a GP, and let $\mathbf{I} = [i_1, i_2, \cdots, i_N]^T \in \mathcal{I}^N$ and $\mathbf{J} = [j_1, j_2, \cdots, i_M]^T \in \mathcal{I}^M$ be two vectors of indices. We define the *associated mean vector* to $\mathbf{I}$, $\mathbf{M}(\mathbf{I})$, and the *associated covariance matrix* to $\mathbf{I}$ and $\mathbf{J}$, $\mathbf{K}(\mathbf{I}, \mathbf{J})$, as

$$\mathbf{M}(\mathbf{I}) = \begin{bmatrix} m(i_1) \\ m(i_2) \\ \vdots \\ m(i_N) \end{bmatrix} \tag{4.2a}$$

$$\mathbf{K}(\mathbf{I}, \mathbf{J}) = \begin{bmatrix} k(i_1,j_1) & k(i_1,j_2) & \cdots & k(i_1,j_M) \\ k(i_2,j_2) & k(i_2,j_2) & \cdots & k(i_2,j_M) \\ \vdots & \vdots & \ddots & \vdots \\ k(i_N,j_1) & k(i_N,j_2) & \cdots & k(i_N,j_M) \end{bmatrix}. \tag{4.2b}$$

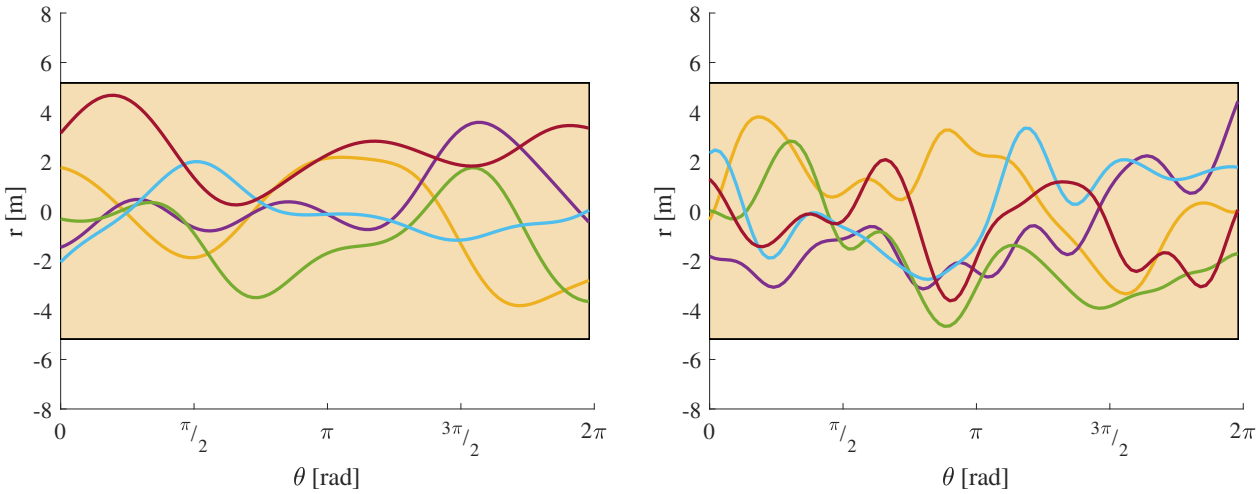
4.1.1 Real processes with squared exponential covariance functions

Since we are interested in estimating the values of an unknown radius function, the most relevant class of GPes in this regard, is the one given by real processes of the form $r(\theta) \sim \mathcal{GP}(m(\theta), k(\theta, \theta'))$, where $\theta, \theta' \in [0, 2\pi]$.

A well-used model for real processes is to choose a zero mean function and a *squared exponential covariance function* (see [23, ch. 1]), which is defined as

$$\tilde{k}_e(\theta, \theta') = \sigma_f^2 e^{-\frac{(\theta-\theta')^2}{2l^2}}, \tag{4.3}$$

where the hyperparameters $\sigma_f > 0$ and $l > 0$ are called the *variance of the signal amplitude* and the *lengthscale*, respectively. The hyperparameter σ_f models how large the variance and covariance of the different function values are, while the length-scale l models how far points have to be so that their function values are uncorrelated.



(a) $l = \frac{\pi}{4}$. **(b)** $l = \frac{\pi}{8}$.

Figure 4.1: Five functions drawn from $\mathcal{GP}(0, \tilde{k}_e)$ with $\sigma_f = 2\,\mathrm{m}$ and different values for l. The interval $[0, 2\pi)$ was regularly discretized using 100 test angles. The shadowed region corresponds to a 99% confidence interval for each function value.

Figure 4.1 shows what type of functions can be modeled by the GP family $\mathcal{GP}(0, \tilde{k}_e)$. Here, the interval $[0, 2\pi)$ was regularly discretized using 100 test angles: $\theta_{t,n} = \frac{2\pi}{N_t}(n-1)$, $N_t = 100$; and five functions were drawn from the resulting multivariate Gaussian distribution $\mathcal{N}(0, \tilde{\mathbf{K}}_e(\mathbf{I}, \mathbf{I}))$, where $\mathbf{I} = [\theta_{t,n}]_{n=1}^{N_t}$. The hyperparameter value $\sigma_f = 2\,\mathrm{m}$ was chosen and two different length-scale values $l = \frac{\pi}{4}$ and $l = \frac{\pi}{8}$ were considered.

Figure 4.1a and Figure 4.1b show the results for $l = \frac{\pi}{4}$ and $l = \frac{\pi}{8}$, respectively. Since $\tilde{k}_e > 0$, there is always a non-zero covariance between two different points, and the resulting functions are therefore smooth. Moreover, it follows from the exponential term in (4.3) that the larger the length-scale l is, the more the covariance between points increases. Hence, functions are more likely to be slowly varying for larger values of l. This can be verified by comparing the functions shown in Figure 4.1a and Figure 4.1b.

As illustrated in Figure 4.1, the covariance function $\tilde{k}_e$ models functions have a mean very close to zero. This effect is undesirable for a radius function, which only takes non-negative values. Another undesirable property of the functions obtained using $\tilde{k}_e$ is their overall smoothness because the unknown radius function may present edges, as is the case of the radius functions presented in Section 2.3. Therefore, we model the unknown radius function as the sum of a function from $\mathcal{GP}(0, \tilde{k}_e)$, an unknown constant value and random white noise:

$$r(\theta) = \tilde{r}(\theta) + r_0 + \eta(\theta) \tag{4.4a}$$
$$\tilde{r}(\theta) \sim \mathcal{GP}(0, \tilde{k}_e(\theta, \theta')) \tag{4.4b}$$
$$r_0 \sim \mathcal{N}(0, \sigma_b^2) \tag{4.4c}$$
$$w(\theta) \sim \mathcal{GP}(0, \sigma_n^2 \delta(\theta, \theta'))\,, \tag{4.4d}$$

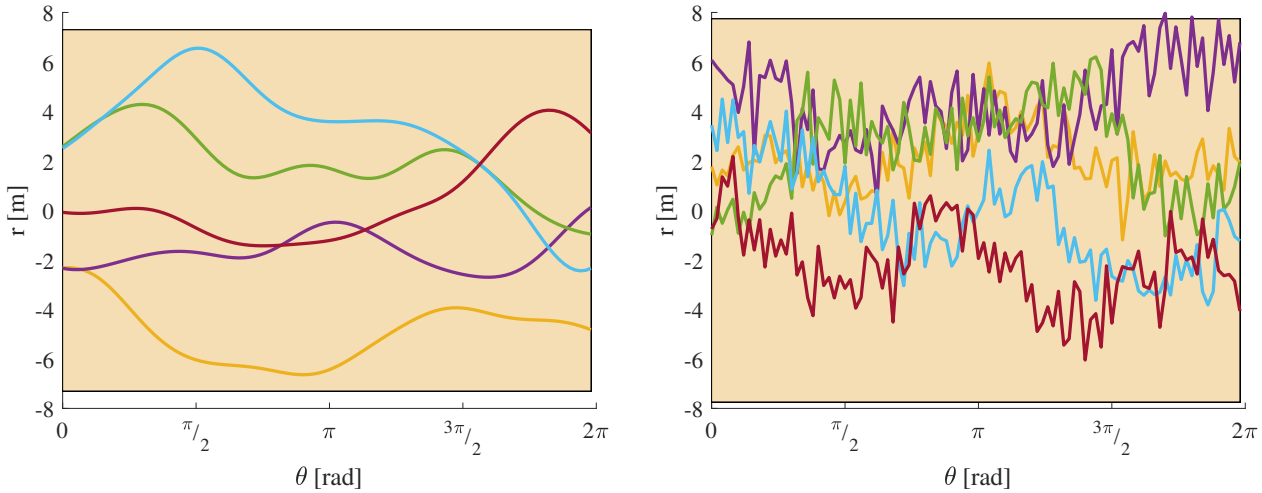
where $\delta(\theta, \theta') = 1$ if $\theta = \theta'$, and $\delta(\theta, \theta') = 0$ otherwise. Under the assumption that $\tilde{r}(\theta)$, r_0 and w are independent of each other, it follows that

$$r(\theta) \sim \mathcal{GP}(0, \tilde{k}_e(\theta, \theta') + \sigma_b^2 + \sigma_n^2 \delta(\theta, \theta')). \tag{4.5}$$

Therefore, we define the GP family $\mathcal{GP}(0, k_e(\theta, \theta'))$, where

$$k_e(\theta, \theta') = \tilde{k}_e(\theta, \theta') + \sigma_b^2 + \sigma_n^2 \delta(\theta, \theta') \tag{4.6a}$$
$$= \sigma_f^2 e^{-\frac{(\theta-\theta')^2}{2l^2}} + \sigma_b^2 + \sigma_n^2 \delta(\theta, \theta') \tag{4.6b}$$

is the *squared exponential covariance function with a bias and a noise term*. The new hyperparameters $\sigma_b \geq 0$ and $\sigma_n \geq 0$ are called the *variance of the bias* and *variance of the white noise*, respectively.



(a) $\sigma_n = 0\,\mathrm{m}$. **(b)** $\sigma_n = 1\,\mathrm{m}$

Figure 4.2: Five functions drawn from $\mathcal{GP}(0, k_e)$ with $\sigma_f = 2\,\mathrm{m}$, $\sigma_b = 2\,\mathrm{m}$, $l = \frac{\pi}{4}$ and different values for σ_n. The interval $[0, 2\pi)$ was regularly discretized using 100 test angles. The shadowed region corresponds to a 99% confidence interval for each function value.

The effects that the hyperparameters σ_b and σ_n have in the resulting GP are illustrated in Figure 4.2. Here, the same test angles used to discretize the interval $[0, 2\pi)$ in the example shown in Figure 4.1 were used, and five functions were drawn from the resulting multivariate

Gaussian distribution. The hyperparameter values $\sigma_f = 2\,\mathrm{m}$, $\sigma_b = 2\,\mathrm{m}$ and $l = \frac{\pi}{4}$ were chosen and two different values for σ_n, $0\,\mathrm{m}$ and $1\,\mathrm{m}$, were considered.

Figure 4.2a and Figure 4.2b illustrate the effect of the hyperparameter σ_b. The functions shown in these figures can have very different mean values compared to the functions shown in Figure 4.1a and Figure 4.1b, whose mean values are very close to zero. On the other hand, Figure 4.2b illustrates the effect of the hyperparameter σ_n, which is to add white random noise to the drawn functions.

The parameter σ_n plays also another important role from a numerical point of view. As discussed in Chapter 2, the unknown radius function is represented by its values at a discretization of $[0, 2\pi)$, $\mathbf{I} = [\theta_t]_{t=1}^{N_t}$. However, if $\sigma_n = 0$, the associated covariance matrix, $\mathbf{K}_e(\mathbf{I}, \mathbf{I})$, may be singular or ill-conditioned since its columns are too similar to each other. This is a major problem for GP regression, which requires the inverse of $\mathbf{K}_e(\mathbf{I}, \mathbf{I})$, as will be shown in Section 4.2. The addition of a white random noise term into the covariance function k_e is equivalent to regularize the matrix $\mathbf{K}_e(\mathbf{I}, \mathbf{I})$ by adding to it the matrix $\sigma_n^2 \mathbf{I}_{N_t}$.

Singular or ill-conditioned covariance matrices $\mathbf{K}(\mathbf{I}, \mathbf{I})$ are a recurrent issue in GP models, and addition of a positive multiple of the identity matrix, also known as a "nugget", is a well-used regularization technique [22].

4.1.2 Real processes for periodic and symmetric functions

As shown in the examples in Section 4.1.1, the covariance function k_e allows us to model a wide variety of different radius functions. The cartesian curves associated to these radius functions may or may not be closed, since the radius curves are not necessarily 2π-periodic. Since the boundary of the object's extent is by definition closed (see Section 2.2), it is desirable to add this condition into the GP model. Furthermore, in some applications, the extent may possess some kind of symmetry, which is known beforehand or that may be considered a reasonable assumption. For example, if the object's extent is the hull of a maritime vessel, it can be assumed that the extent is symmetric about its longitudinal axis. Such knowledge about the symmetries of the extent may drastically improve the estimation of the extent. Therefore, it is also desirable to add a priori knowledge about the symmetries of the extent into the GP that models the unknown radius function.

In this section, the covariance functions $\tilde{k}_e$ and k_e will be modified to model 2π-periodic, π-periodic and even radius functions. 2π-periodic radius functions correspond to closed cartesian curves. If the radius function is in addition π-periodic or even, then the closed curve is symmetry about the center or symmetry about the x-axis, respectively.

2π-periodic radius functions are modeled by replacing $\tilde{k}_e$ with

$$\tilde{k}_{2\pi}(\theta, \theta') = \sigma_f^2 e^{-\frac{2}{l^2}\sin^2\left(\frac{\theta-\theta'}{2}\right)}; \tag{4.7}$$

π-periodic radius functions are modeled by replacing $\tilde{k}_e$ with

$$\tilde{k}_{\pi}(\theta, \theta') = \sigma_f^2 e^{-\frac{1}{2l^2}\sin^2(\theta-\theta')}; \tag{4.8}$$

and 2π-periodic even radius functions are modeled by replacing $\tilde{k}_e$ with

$$\tilde{k}_{2\pi,a}(\theta,\theta') = \sigma_f^2 e^{-\frac{1}{2l^2}(|\operatorname{ssa}(\theta)|-|\operatorname{ssa}(\theta')|)^2}, \tag{4.9}$$

where $\operatorname{ssa}(\cdot)$ is the *smallest signed angle function*, which is defined as

$$\operatorname{ssa}(\theta) = \pi - [(\pi-\theta) \pmod{2\pi}], \tag{4.10}$$

i.e. $\operatorname{ssa}(\theta)$ is the only angle in $(-\pi,\pi]$ such that $\operatorname{ssa}(\theta) \equiv \theta \pmod{2\pi}$.

Similarly as done with $\tilde{k}_e$, we add bias and noise terms to $\tilde{k}_{2\pi}$, $\tilde{k}_\pi$ and $\tilde{k}_{2\pi,a}$. This yields the following covariance functions

$$k_{2\pi}(\theta,\theta') = \sigma_f^2 e^{-\frac{2}{l^2}\sin^2\left(\frac{\theta-\theta'}{2}\right)} + \sigma_b^2 + \sigma_n^2\delta(\theta,\theta') \tag{4.11a}$$

$$k_{\pi}(\theta,\theta') = \sigma_f^2 e^{-\frac{1}{2l^2}\sin^2(\theta-\theta')} + \sigma_b^2 + \sigma_n^2\delta(\theta,\theta') \tag{4.11b}$$

$$k_{2\pi,a}(\theta,\theta') = \sigma_f^2 e^{-\frac{1}{2l^2}(|\operatorname{ssa}(\theta)|-|\operatorname{ssa}(\theta')|)^2} + \sigma_b^2 + \sigma_n^2\delta(\theta,\theta'). \tag{4.11c}$$

The same observations done in Section 4.1.1 about the effect that the hyperparameters σ_f, σ_b, σ_n and l have on the functions modeled by $\mathcal{GP}(0,k_e)$, also apply for the GPes $\mathcal{GP}(0,k_{2\pi})$, $\mathcal{GP}(0,k_\pi)$ and $\mathcal{GP}(0,k_{2\pi,a})$.

As a consequence of the definition of these covariance functions, we obtain the correlations

$$\rho_{2\pi}(\theta,\theta+2\pi) = \frac{k_{2\pi}(\theta,\theta+2\pi)}{\sqrt{k_{2\pi}(\theta,\theta)}\sqrt{k_{2\pi}(\theta+2\pi,\theta+2\pi)}} = \frac{\sigma_f^2+\sigma_b^2}{\sigma_f^2+\sigma_b^2+\sigma_n^2} \tag{4.12a}$$

$$\rho_{\pi}(\theta,\theta+\pi) = \frac{k_{\pi}(\theta,\theta+\pi)}{\sqrt{k_{\pi}(\theta,\theta)}\sqrt{k_{\pi}(\theta+\pi,\theta+\pi)}} = \frac{\sigma_f^2+\sigma_b^2}{\sigma_f^2+\sigma_b^2+\sigma_n^2} \tag{4.12b}$$

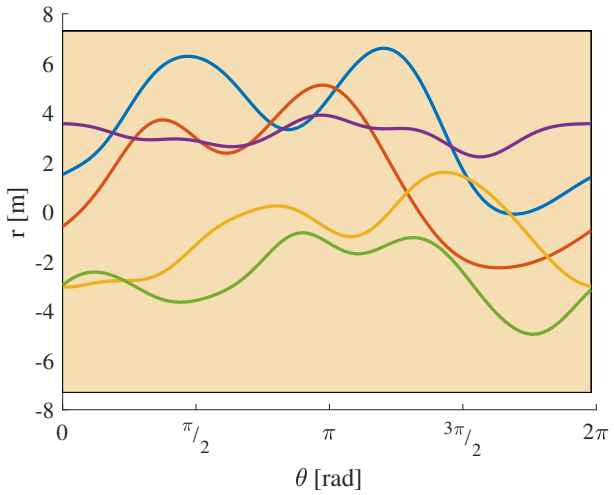
$$\rho_{2\pi,a}(\theta,\theta+2\pi) = \frac{k_{2\pi,a}(\theta,\theta+2\pi)}{\sqrt{k_{2\pi,a}(\theta,\theta)}\sqrt{k_{2\pi,a}(\theta+2\pi,\theta+2\pi)}} = \frac{\sigma_f^2+\sigma_b^2}{\sigma_f^2+\sigma_b^2+\sigma_n^2} \tag{4.12c}$$

$$\rho_{2\pi,a}(\theta,-\theta) = \frac{k_{2\pi,a}(\theta,-\theta)}{\sqrt{k_{2\pi,a}(\theta,\theta)}\sqrt{k_{2\pi,a}(-\theta,-\theta)}} = \frac{\sigma_f^2+\sigma_b^2}{\sigma_f^2+\sigma_b^2+\sigma_n^2}. \tag{4.12d}$$

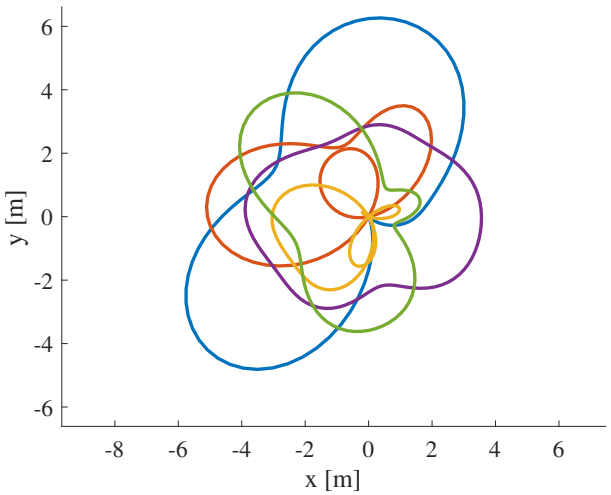
$$\tag{4.12e}$$

Therefore, if there is no white random noise term, i.e. $\sigma_n = 0$, all these correlations are equal to 1. In other words, if no white random noise is added, $\mathcal{GP}(0,k_{2\pi})$ models radius functions that are 2π-periodic, $\mathcal{GP}(0,k_\pi)$ models radius functions that are π-periodic, and $\mathcal{GP}(0,k_{2\pi,a})$ models radius functions that are 2π-periodic and even.

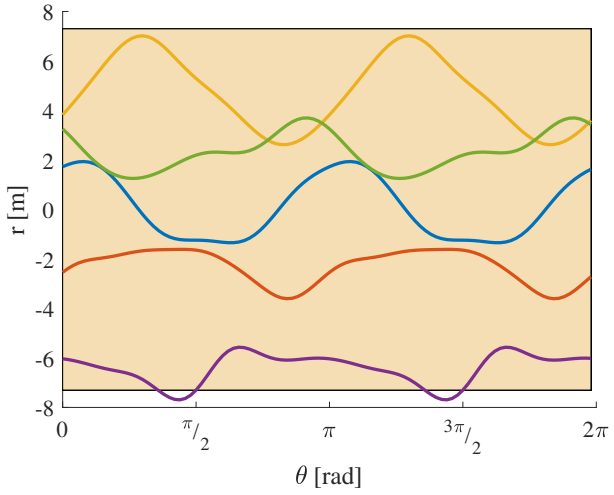
On the other hand, if there is a white random noise term, i.e. $\sigma_n > 0$, then these correlations are less than 1, and the modeled radius functions do not necessarily satisfy the above-mentioned properties, which is as expected due to the nature of white random noise.



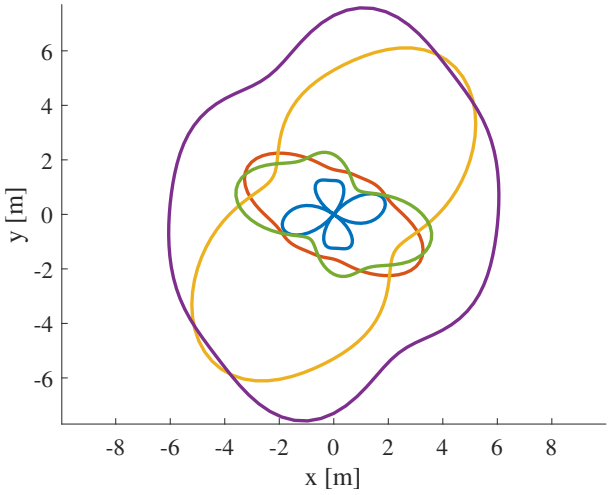
(a) Radius functions drawn for $k = k_{2\pi}$.



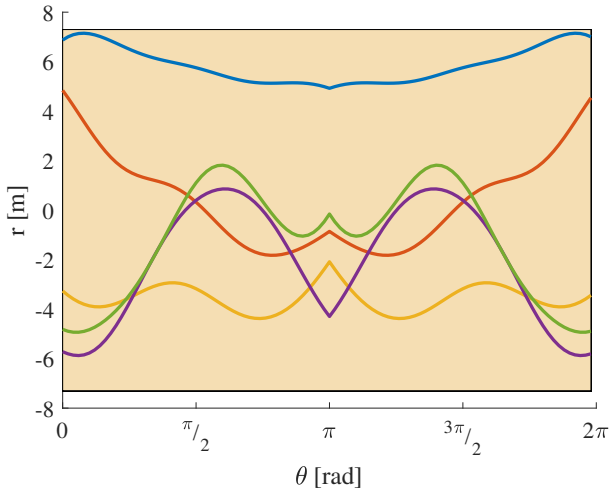
(b) Cartesian curves obtained for $k = k_{2\pi}$.



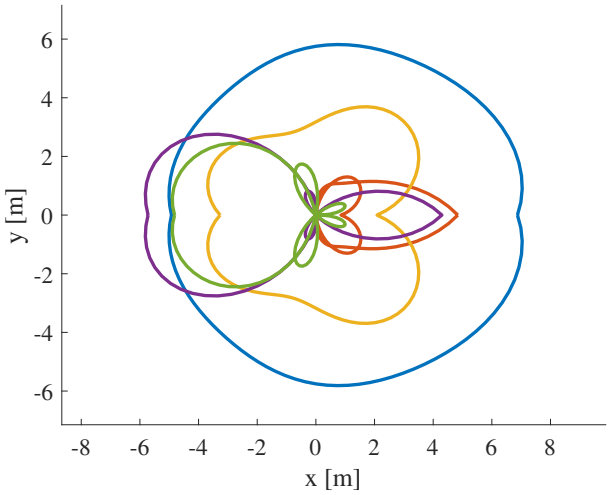
(c) Radius functions drawn for $k = k_{\pi}$.



(d) Cartesian curves obtained for $k = k_{\pi}$.



(e) Radius functions drawn for $k = k_{2\pi,a}$.



(f) Cartesian curves obtained for $k = k_{2\pi,a}$.

Figure 4.3: Five radius functions drawn from $\mathcal{GP}(0, k)$ with $(\sigma_f, \sigma_b, \sigma_n, l) = (2\,\mathrm{m}, 2\,\mathrm{m}, 0\,\mathrm{m}, \frac{\pi}{4})$ for $k = k_{2\pi}$, $k = k_{\pi}$ and $k = k_{2\pi,a}$, and their associated cartesian curves. The interval $[0, 2\pi)$ was regularly discretized using 100 test angles. The shadowed region corresponds to a 99% confidence interval for each function value.

Figure 4.3 illustrates the radius functions modeled by the GP families $\mathcal{GP}(0, k_{2\pi})$, $\mathcal{GP}(0, k_{\pi})$ and $\mathcal{GP}(0, k_{2\pi,a})$ for $\sigma_n = 0$, as well as their associated cartesian curves. As done in the examples shown in Figure 4.1 and Figure 4.2, the interval $[0, 2\pi)$ was regularly discretized using 100 test angles, and five radius functions were drawn from the resulting multivariate Gaussian distribution for each covariance function. The hyperparameters used were $(\sigma_f, \sigma_b, \sigma_n, l) = (2\,\mathrm{m}, 2\,\mathrm{m}, 0\,\mathrm{m}, \frac{\pi}{4})$.

The cartesian curves generated by the drawn radius functions are continuous and closed (see Figures 4.3b, 4.3d and 4.3f). The cartesian curves obtained using $k_{2\pi}$ (Figure 4.3b) do not posses any particular type of symmetry, while the cartesian curves obtained using k_{π} (Figure 4.3d) and $k_{2\pi,a}$ (Figure 4.3f) are symmetric about the center and the x-axis, respectively.

The scaling factors in the exponents of $\tilde{k}_{2\pi}$, $\tilde{k}_{\pi}$ and $\tilde{k}_{2\pi,a}$ are chosen such that $k_{2\pi}(\theta, \theta')$, $k_{\pi}(\theta, \theta')$ and $k_{2\pi,a}(\theta, \theta')$ are very similar to $k_e(\theta, \theta')$, when $\theta' \to \theta$. This is a consequence of the limits

$$\lim_{\theta' \to \theta} \frac{\ln(\tilde{k}_{2\pi}(\theta, \theta'))}{\ln(\tilde{k}_e(\theta, \theta'))} = \lim_{\theta' \to \theta} \frac{-\frac{2}{l^2}\sin^2\left(\frac{\theta - \theta'}{2}\right)}{-\frac{(\theta-\theta')^2}{2l^2}} = \lim_{\theta' \to \theta} \left(\frac{\sin\left(\frac{\theta-\theta'}{2}\right)}{\frac{\theta-\theta'}{2}}\right)^2 = 1 \tag{4.13a}$$

$$\lim_{\theta' \to \theta} \frac{\ln(\tilde{k}_{\pi}(\theta, \theta'))}{\ln(\tilde{k}_e(\theta, \theta'))} = \lim_{\theta' \to \theta} \frac{-\frac{1}{2l^2}\sin^2(\theta - \theta')}{-\frac{(\theta-\theta')^2}{2l^2}} = \lim_{\theta' \to \theta} \left(\frac{\sin(\theta - \theta')}{\theta - \theta'}\right)^2 = 1 \tag{4.13b}$$

$$\lim_{\theta' \to \theta} \frac{\ln(\tilde{k}_{2\pi,a}(\theta, \theta'))}{\ln(\tilde{k}_e(\theta, \theta'))} = \lim_{\theta' \to \theta} \frac{-\frac{2}{l^2}\left(|\operatorname{ssa}(\theta)| - |\operatorname{ssa}(\theta')|\right)^2}{-\frac{(\theta-\theta')^2}{2l^2}} \tag{4.13c}$$

$$= \lim_{\theta' \to \theta} \left(\frac{|\operatorname{ssa}(\theta)| - |\operatorname{ssa}(\theta')|}{\theta - \theta'}\right)^2 = 1\,, \tag{4.13d}$$

where the last limit follows from $|\operatorname{ssa}(\cdot)|$ being a saw-tooth function as shown in Figure 4.4.

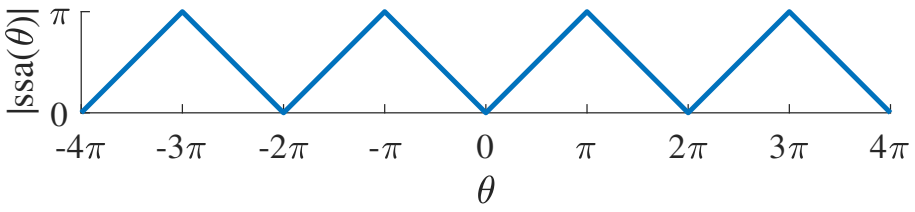


Figure 4.4: $|\operatorname{ssa}(\theta)|$ for $\theta \in [-4\pi, 4\pi]$.

The limits in Equation (4.13) imply that $k_{2\pi}(\theta, \theta'), k_{\pi}(\theta, \theta'), k_{2\pi,a}(\theta, \theta') \to k_e(\theta, \theta')$ exponentially when $\theta' \to \theta$. Therefore, we conclude that radius functions modeled by GPes that use the covariance function $k_{2\pi}$, k_{π} or $k_{2\pi}$ are locally similar to radius functions modeled by GPes that use the covariance function k_e.

The covariance functions k_e, $k_{2\pi}$ and k_{π} can be found in the literature. See for example [23] and [28]. However, the covariance function $k_{2\pi,a}$ or similar is not present in the literature to the best of the authors knowledge, and this covariance function was explicitly developed for this study in order to model axial symmetry.

4.2 Gaussian process regression

GP regression is a method for estimating the values of an unknown function that is modeled as a real GP based on noisy measurements of some of the function values. In this section, we present the theoretical background and important equations for this regression method, as well as an example that involves some of the covariance functions defined in Section 4.1.

Let $f(x) \sim \mathcal{GP}(m(x), k(x, x'))$ be a real GP, where $x, x' \in \mathcal{I}$ and $\mathcal{I} \subset \mathbb{R}$ is an interval, and let $\mathbf{x}_t = [x_{t,1}, x_{t,2}, \cdots, x_{t,N_t}]^T = [x_t]_{t=1}^{N_t} \in \mathcal{I}^{N_t}$ be a vector of different points in $\mathcal{I}$, which are called *test points*. The test points are points at which the function values are of special interest. In the case of the radius function that describes the object's extent, the test points are the test angles $\boldsymbol{\theta}_t = [\theta_t]_{t=1}^{N_t}$ defined in Chapter 2, which are used to discretize the radius function.

The measurements $\mathbf{y} = [y_1, y_2, \cdots, y_{N_i}] = [y_n]_{n=1}^{N_i} \in \mathbb{R}^{N_i}$ of the function values at the points $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,N_i}] = [x_{i,n}]_{n=1}^{N_i} \in \mathcal{I}^{N_i}$ are modeled as

$$y_n = f(x_{i,n}) + w_n \,, \tag{4.14}$$

where $w_n \sim \mathcal{N}(0, \sigma_w^2)$ is the measurement noise and $\sigma_w > 0$ is the measurement noise strength. The noise at each measurement is assumed independent of each other, and independent of the function values.

The points at which the function values are measured, $\mathbf{x}_i = [x_{i,n}]_{n=1}^{N_i}$ are called *input points*. The denominations "test" and "input" points have their origin in the Machine Learning field (see [23, ch. 2]).

The GP model and the independence assumption on the measurement noise yields that

$$\begin{bmatrix} \mathbf{f}_i \\ \mathbf{f}_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{M}(\mathbf{x}_i) \\ \mathbf{M}(\mathbf{x}_t) \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{x}_i, \mathbf{x}_i) & \mathbf{K}(\mathbf{x}_i, \mathbf{x}_t) \\ \mathbf{K}(\mathbf{x}_t, \mathbf{x}_i) & \mathbf{K}(\mathbf{x}_t, \mathbf{x}_t) \end{bmatrix} \right) \tag{4.15a}$$

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{M}(\mathbf{x}_i) \\ \mathbf{M}(\mathbf{x}_t) \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{x}_i, \mathbf{x}_i) + \sigma_w^2 \mathbf{I}_{N_i} & \mathbf{K}(\mathbf{x}_i, \mathbf{x}_t) \\ \mathbf{K}(\mathbf{x}_t, \mathbf{x}_i) & \mathbf{K}(\mathbf{x}_t, \mathbf{x}_t) \end{bmatrix} \right) , \tag{4.15b}$$

where $\mathbf{f}_i = [f(x_{i,1}), f(x_{i,2}), \cdots f(x_{i,N_i})]^T \in \mathbb{R}^{N_i}$ and $\mathbf{f}_t = [f(x_{t,1}), f(x_{t,2}), \cdots f(x_{t,N_t})]^T \in \mathbb{R}^{N_t}$ are the function values at the input and test points, respectively.

By using the result from Appendix A.1, we conclude by that the conditional distributions $\mathbf{f}_t|\mathbf{y}$, $\mathbf{y}|\mathbf{f}_t$ and $\mathbf{f}_i|\mathbf{f}_t$ are normally distributed with mean vectors and variances given by

$$\mathbb{E}[\mathbf{f}_t|\mathbf{y}] = \mathbf{M}(\mathbf{x}_t) + \mathbf{K}(\mathbf{x}_t, \mathbf{x}_i)[\mathbf{K}(\mathbf{x}_i, \mathbf{x}_i) + \sigma_w^2 \mathbf{I}_{N_i}]^{-1} (\mathbf{y} - \mathbf{M}(\mathbf{x}_i)) \tag{4.16a}$$

$$\mathbb{V}[\mathbf{f}_t|\mathbf{y}] = \mathbf{K}(\mathbf{x}_t, \mathbf{x}_t) - \mathbf{K}(\mathbf{x}_t, \mathbf{x}_i)[\mathbf{K}(\mathbf{x}_i, \mathbf{x}_i) + \sigma_w^2 \mathbf{I}_{N_i}]^{-1} \mathbf{K}[\mathbf{x}_i, \mathbf{x}_t] \tag{4.16b}$$

$$\mathbb{E}[\mathbf{y}|\mathbf{f}_t] = \mathbf{M}(\mathbf{x}_i) + \mathbf{K}(\mathbf{x}_i, \mathbf{x}_t)\mathbf{K}(\mathbf{x}_t, \mathbf{x}_t)^{-1} (\mathbf{f}_t - \mathbf{M}(\mathbf{x}_t)) \tag{4.16c}$$

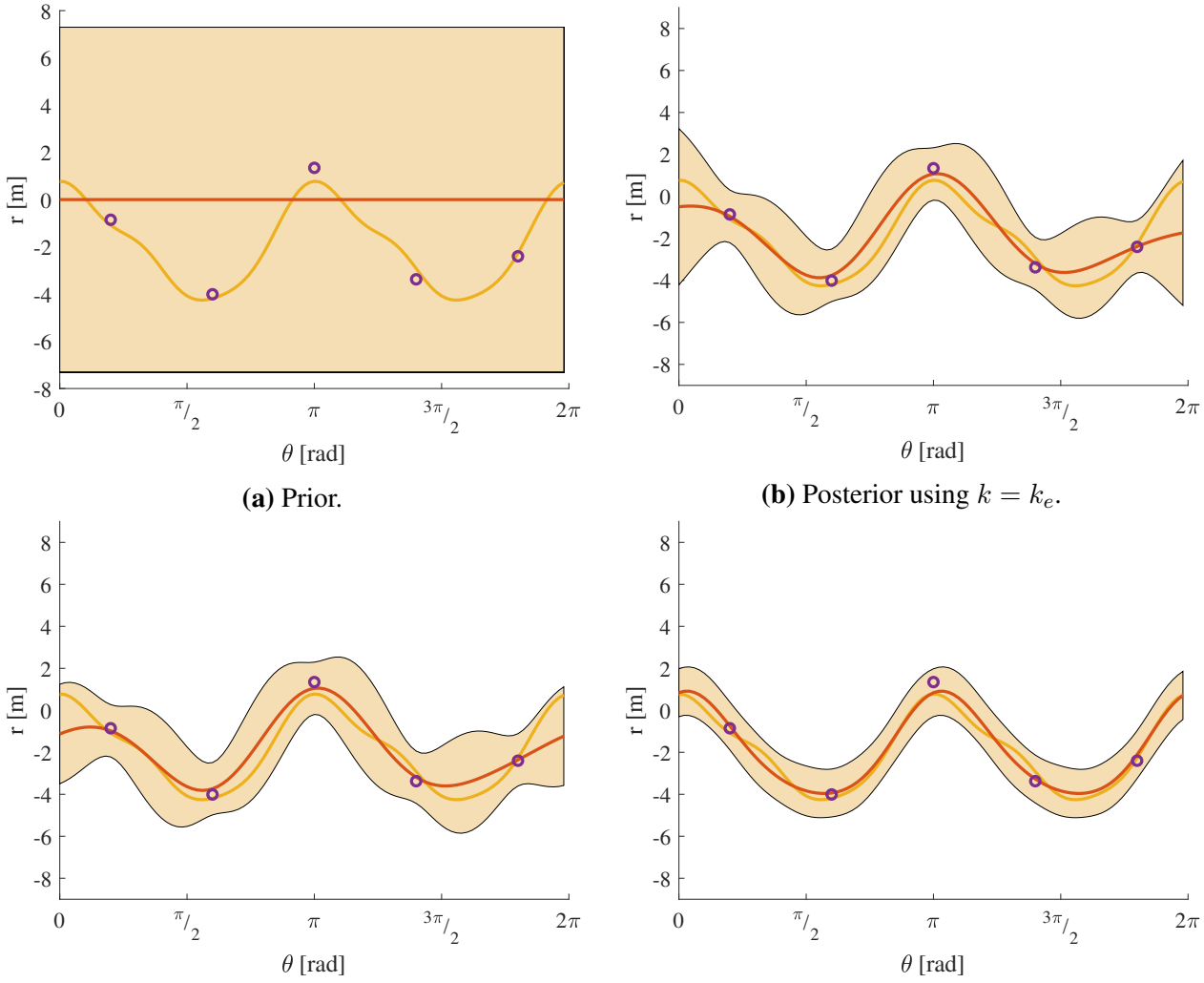
$$\mathbb{V}[\mathbf{y}|\mathbf{f}_t] = \mathbf{K}(\mathbf{x}_i, \mathbf{x}_i) + \sigma_w^2 \mathbf{I}_{N_i} - \mathbf{K}(\mathbf{x}_i, \mathbf{x}_t)\mathbf{K}(\mathbf{x}_t, \mathbf{x}_t)^{-1} \mathbf{K}[\mathbf{x}_t, \mathbf{x}_i] \tag{4.16d}$$

$$\mathbb{E}[\mathbf{f}_i|\mathbf{f}_t] = \mathbf{M}(\mathbf{x}_i) + \mathbf{K}(\mathbf{x}_i, \mathbf{x}_t)\mathbf{K}(\mathbf{x}_t, \mathbf{x}_t)^{-1} (\mathbf{f}_t - \mathbf{M}(\mathbf{x}_t)) \tag{4.16e}$$

$$\mathbb{V}[\mathbf{f}_i|\mathbf{f}_t] = \mathbf{K}(\mathbf{x}_i, \mathbf{x}_i) - \mathbf{K}(\mathbf{x}_i, \mathbf{x}_t)\mathbf{K}(\mathbf{x}_t, \mathbf{x}_t)^{-1} \mathbf{K}[\mathbf{x}_t, \mathbf{x}_i] \tag{4.16f}$$

Equations (4.16a) to (4.16b) summarize GP regression: The prior distribution of the function values at the test points is $\mathbf{f}_t \sim \mathcal{N}(\mathbf{M}(\mathbf{x}_t), \mathbf{K}(\mathbf{x}_t, \mathbf{x}_t))$ due to the GP model assumption,

and given the measurements $\mathbf{y}$, the posterior distribution of these function values, $\mathbf{f}_t|\mathbf{y}$, is normally distributed with mean vector and covariance matrix given by Equation (4.16a) and Equation (4.16b), respectively. Furthermore, Equations (4.16c) to (4.16d) provides the likelihood of the measurements given the function values at the test points, while Equations (4.16e) to (4.16f) may be used to estimate function values at arbitrary points based on the estimates at the test points.



(a) Prior.

(b) Posterior using $k = k_e$.

(c) Posterior using $k = k_{2\pi}$.

(d) Posterior using $k = k_\pi$.

Figure 4.5: GP regression: True function (yellow), measurements (purple), mean of $\mathbf{f}_t$ (red) and a 99% confidence region for $\mathbf{f}_t$ (shadowed region). Hyperparameter values $(\sigma_f, \sigma_b, \sigma_n, l) = (2\,\mathrm{m}, 2\,\mathrm{m}, 0, \frac{\pi}{4})$.

Figure 4.5 shows an GP regression example. As done in previous examples (see Section 4.1), the test points correspond to a regular discretization of the interval $[0, 2\pi)$ of 100 points. The true function to be estimated is drawn from $\mathcal{GP}(0, k_\pi)$ with hyperparameters $(\sigma_f, \sigma_b, \sigma_n, l) = (2\,\mathrm{m}, 2\,\mathrm{m}, 0, \frac{\pi}{4})$, and 5 measurements are taken with $\sigma_w = 0.5\,\mathrm{m}$ at input points on $[0, 2\pi]$ that are equidistant to each other. GP regression is performed using $\mathcal{GP}(0, k_e)$, $\mathcal{GP}(0, k_{2\pi})$ and $\mathcal{GP}(0, k_\pi)$.

Figure 4.5a shows the prior distribution of $\mathbf{f}_t$: the mean vector (red) and a 99% confidence

interval for each function value (shadowed region) is the same for each considered GP. Figures 4.5b to 4.5d show the GP regression results using $\mathcal{GP}(0, k_e)$, $\mathcal{GP}(0, k_{2\pi})$ and $\mathcal{GP}(0, k_\pi)$, respectively. In each case, the mean vector of the posterior distribution $\mathbf{f}_t|\mathbf{y}$ is close to the true function values, and the true function lies inside the 99% confidence region for the estimated function values, which is considerably narrower than the one for the prior distribution. The results using the covariance function $k_{2\pi}$ are better than the ones obtained using k_e since $\mathcal{GP}(0, k_{2\pi})$ correctly assumes that the true function is 2π-periodic, and uses the measurements to estimate the function values at test points that are 2π units away of the input points. This explains why the estimate at test points close to 0 and 2π are considerably better. By a similar argument, we can explain why the estimates using k_π are even better than the ones obtained using $k_{2\pi}$.

Equations (4.16a) to (4.16f) are used extensively in Chapter 5 together with the GPes defined in section 4.1.2, which have zero mean function, i.e. $m(x) = 0$. Therefore, by defining the notations

$$\mathbf{H}_{\mathcal{GP}}(\mathbf{x}_1, \mathbf{x}_2, \sigma) = \mathbf{K}(\mathbf{x}_1, \mathbf{x}_2)[\mathbf{K}(\mathbf{x}_2, \mathbf{x}_2) + \sigma^2 \mathbf{I}_{N_i}]^{-1} \tag{4.17a}$$

$$\mathbf{R}_{\mathcal{GP}}(\mathbf{x}_1, \mathbf{x}_2, \sigma) = \mathbf{K}(\mathbf{x}_1, \mathbf{x}_1) - \mathbf{K}(\mathbf{x}_1, \mathbf{x}_2)[\mathbf{K}(\mathbf{x}_2, \mathbf{x}_2) + \sigma^2 \mathbf{I}_{N_i}]^{-1}\mathbf{K}(\mathbf{x}_2, \mathbf{x}_1)\,, \tag{4.17b}$$

we can rewrite the conditional distributions for $\mathbf{f}_t|\mathbf{y}$, $\mathbf{y}|\mathbf{f}_t$ and $\mathbf{f}_i|\mathbf{f}_t$ when the mean function is zero as

$$\mathbf{f}_t|\mathbf{y} \sim \mathcal{N}\left(\mathbf{H}_{\mathcal{GP}}(\mathbf{x}_t, \mathbf{x}_i, \sigma_w)\mathbf{y}, \mathbf{R}_{\mathcal{GP}}(\mathbf{x}_t, \mathbf{x}_i, \sigma_w)\right) \tag{4.18a}$$

$$\mathbf{y}|\mathbf{f}_t \sim \mathcal{N}\left(\mathbf{H}_{\mathcal{GP}}(\mathbf{x}_i, \mathbf{x}_t, 0)\mathbf{f}_t, \mathbf{R}_{\mathcal{GP}}(\mathbf{x}_i, \mathbf{x}_t, 0) + \sigma_w^2 \mathbf{I}_{N_i}\right) \tag{4.18b}$$

$$\mathbf{f}_i|\mathbf{f}_t \sim \mathcal{N}\left(\mathbf{H}_{\mathcal{GP}}(\mathbf{x}_i, \mathbf{x}_t, 0)\mathbf{f}_t, \mathbf{R}_{\mathcal{GP}}(\mathbf{x}_i, \mathbf{x}_t, 0)\right) \tag{4.18c}$$

4.3 Model selection: GP model and hyperparameter values

Model selection is the problem of choosing a GP model for a particular application, i.e. choosing a mean and covariance function, as well as determining hyperparameter values. In general, model selection is far from a trivial task. The main reason for this is that GPes are non-parametric models. Therefore it may not be obvious which GP model family or parameter values should be chosen. In addition, some covariance functions may depend on many hyperparameters [23, p.105-106]. However, depending on the particular application, the GP model family or the value of some hyperparameters may be easy to specify.

In the particular case of this study, the GP families $\mathcal{GP}(0, k_{2\pi})$, $\mathcal{GP}(0, k_\pi)$ and $\mathcal{GP}(0, k_{2\pi,a})$ seem to be reasonable choices to model the radius function that parameterizes the boundary of the object's extend, based on the discussion in Section 4.1.2. The hyperparameter values for these GP models are chosen heuristically. The hyperparameters σ_b and σ_f are chosen to resemble the mean of the radius function and the difference between the maximum and minimum of this function, respectively. In particular, σ_b and σ_f give overall properties of the radius function. Since the radius function is unknown, it is not reasonable to assume that these quantities are known. Therefore, we assume that in the target tracking initialization process, upper bound estimates for these quantities are obtained. For the hull parameters in 2.1 the average of the radius function for the "spline", "parabola" and "ellipse" methods is between 3.34-$3.51\,\mathrm{m}$, while the

difference between the maximum and minimum radius value is between 2.75-2.95 m. Therefore, the values $\sigma_b = 3.5\,\mathrm{m}$ and $\sigma_f = 3\,\mathrm{m}$ are chosen. The value of the hyperparameter σ_n is set to $\sigma_n = 0.1\,\mathrm{m}$ since arbitrary variations of the radius function of this scale can be assumed reasonable for the size of the hull. Finally, the lengthscale l is set to $\frac{\pi}{4}$ since $4l = \pi$ and radius values for opposite directions can be assumed uncorrelated.

Hyperparameter	Value
σ_f	$3.0\,\mathrm{m}$
σ_b	$3.5\,\mathrm{m}$
σ_n	$0.1\,\mathrm{m}$
l	$\frac{\pi}{4}$

Table 4.1: Hyperparamaters values for the GPes $\mathcal{GP}(0, k_{2\pi})$, $\mathcal{GP}(0, k_{\pi})$ and $\mathcal{GP}(0, k_{2\pi,a})$.

The chosen hyperparameter values are summarized in table 4.1, and in Chapter 7, it is discussed how the model selection could be improved.

Chapter 5

State-space models and Kalman filters

5.1 Particular case: Known pose

We consider first the case where the pose of the object is known, i.e. the position of the center $\mathbf{r}_c^w = [N_c, E_c]^T$ and heading ψ_c are known. In particular, the linear velocity $[v_N, v_E] = [\dot{N}_c, \dot{E}_c]$ and the angular velocity $r_c = \dot{\psi}_c$ are also available. In other words, the states that describe the kinematics of the object are all known, and only the states that describe the object's extent have to be estimated. Therefore, the state vector for this model is the value of the unknown radius function at the test angles:

$$\mathbf{x} = \mathbf{x}_t = [r(\boldsymbol{\theta}_t)]_{t=1}^{N_t} = \begin{bmatrix} r(\theta_1) & r(\theta_2) & \cdots & r(\theta_{N_t}) \end{bmatrix}^T. \tag{5.1}$$

Since the object's extent is assumed to be a rigid body (see Section 1.3), the state values do not change with time, i.e.

$$\mathbf{x}_k = \mathbf{x}_{k-1} \tag{5.2}$$

In particular, the process model is not subjected to process noise.

We will now show that the knowledge of N_c, E_c and ψ_c allows us to derive a linear measurement model. As discussed in Section 3.2, a lidar range measurement r_m for an azimuth angle φ_m corresponds to a point with coordinates in the world frame given by

$$\mathbf{z}_m^w = \begin{bmatrix} r_m \cos(\varphi_m) \\ r_m \sin(\varphi_m) \end{bmatrix}, \tag{5.3}$$

where we have used that the lidar sensor is placed at the origin of the world frame.

The cartesian coordinates of the measurement, $\mathbf{z}_m^w$, can be related to a radius function measurement y_i at an input angle θ_i using the transformation between the body and world coordinates (eq. (2.2)), which yields

$$\mathbf{z}_m^w = \mathbf{r}_c^w + \begin{bmatrix} \cos(\psi_c) & -\sin(\psi_c) \\ \sin(\psi_c) & \cos(\psi_c) \end{bmatrix} \begin{bmatrix} y_i \cos(\theta_i) \\ y_i \sin(\theta_i) \end{bmatrix} = \mathbf{r}_c^w + y_i \begin{bmatrix} \cos(\theta_i + \psi_c) \\ \sin(\theta_i + \psi_c) \end{bmatrix}, \tag{5.4}$$

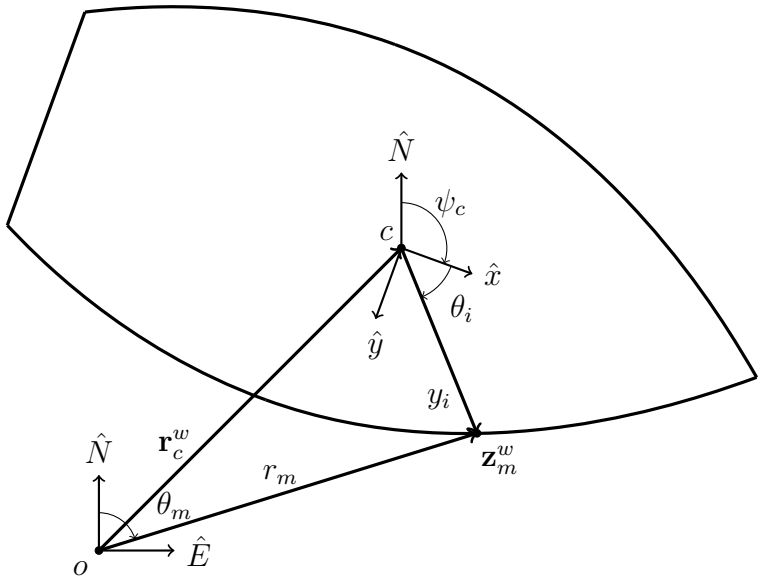


Figure 5.1: Relation between lidar measurements and radius function measurements.

where $\mathbf{r}_c^w = [N_c, E_c]^T$ are the coordinates of the center in the world frame. Hence,

$$r_m \begin{bmatrix} \cos(\varphi_m) \\ \sin(\varphi_m) \end{bmatrix} = \begin{bmatrix} N_c \\ E_c \end{bmatrix} + y_i \begin{bmatrix} \cos(\theta_i + \psi_c) \\ \sin(\theta_i + \psi_c) \end{bmatrix} \tag{5.5a}$$

Therefore, we have the following transformations between the lidar measurement (r_m, φ_m) and the radius function measurements (y_i, θ_i)

$$\begin{cases} y_i &= \left| [r_m \cos(\varphi_m) - N_c, r_m \sin(\varphi_m) - E_c]^T \right| \\ \theta_i &= \angle \left([r_m \cos(\varphi_m) - N_c, r_m \sin(\varphi_m) - E_c]^T \right) - \psi_c \end{cases} \tag{5.6a}$$

$$\begin{cases} r_m &= \left| [N_c + y_i \cos(\theta_i + \psi_c), E_c + y_i \sin(\theta_i + \psi_c)]^T \right| \\ \varphi_m &= \angle \left([N_c + y_i \cos(\theta_i + \psi_c), E_c + y_i \sin(\theta_i + \psi_c)]^T \right) \ , \end{cases} \tag{5.6b}$$

where $|\cdot|$ and $\angle(\cdot)$ are the *norm* and *angle functions*, which are defined as

$$|\mathbf{v}| = \sqrt{v_1^2 + v_2^2} \tag{5.7a}$$

$$\angle(\mathbf{v}) = \operatorname{atan2}(v_2, v_1) = \begin{cases} \arctan\left(\frac{v_2}{v_1}\right), & v_1 > 0 \\ \arctan\left(\frac{v_2}{v_1}\right) + \pi, & v1 < 0, v_2 \geq 0 \\ \arctan\left(\frac{v_2}{v_1}\right) - \pi, & v1 < 0, v_2 < 0 \\ +\frac{\pi}{2}, & v1 = 0, v_2 > 0 \\ -\frac{\pi}{2}, & v1 = 0, v_2 < 0 \end{cases} \tag{5.7b}$$

for $\mathbf{v} = [v_1, v_2]^T \in \mathbb{R}^2 - \{0\}$.

The transformations 5.6a and 5.6b are well-defined as long as $\mathbf{z}_m^w \neq \mathbf{r}_c^w$ and $\mathbf{z}_m^w \neq \mathbf{0}$, respectively. Under the reasonable assumptions that the center lies inside the object's extent and

that the object's extent never reaches the lidar sensor position, the lidar measurements cannot correspond to the center or the origin of the world frame. Therefore, these transformations are well-defined. The relation between (r_m, φ_m), $\mathbf{z}_m^w$ and (y_i, θ_i) is illustrated in Figure 5.1.

The transformations in 5.6a and 5.6b give a one-to-one relation between the lidar measurement (r_m, φ_m) and the radius function measurements (y_i, θ_i), where the angles φ_m and θ_i are unique up to modulo 2π, which does not suppose any inconvenient since these angles are only used as arguments in trigonometric functions. Hence, for a vector $\mathbf{r}_k = [r_{k,1}, r_{k,2}, \cdots, r_{k,N_{k,m}}]^T$ of lidar range measurements at time k for azimuth angles $\boldsymbol{\varphi}_k = [\varphi_{k,1}, \varphi_{k,2}, \cdots, \varphi_{k,N_{k,m}}]^T$, the corresponding radius function measurements $\mathbf{y}_k = [y_{k,1}, y_{k,2}, \cdots, y_{k,N_{k,m}}]^T$ and input angles $\boldsymbol{\theta}_{i,k} = [\theta_{k,1}, \theta_{k,2}, \cdots, \theta_{k,N_{k,m}}]^T$ can be found, and viceversa.

Since the radius function is described by a GP model, the likelihood of the measurements $\mathbf{y}_k = [y_{k,m}]_{m=1}^{N_{k,m}}$ given the values of the radius function at the test angle $\mathbf{x}_t = [r(\theta_{t,n})]_{n=1}^{N_t}$ is given by

$$\mathbf{y}_k | \mathbf{x}_t \sim \mathcal{N}\left(\mathbf{H}_{\mathcal{GP}}(\boldsymbol{\theta}_{i,k}, \boldsymbol{\theta}_t, 0)\mathbf{x}_t, \mathbf{R}_{\mathcal{GP}}(\boldsymbol{\theta}_{i,k}, \boldsymbol{\theta}_t, 0) + \sigma_r^2 \mathbf{I}_{N_{k,m}}\right) \tag{5.8}$$

according to eq. (4.18b), where σ_r is the range measurement noise strength of the lidar sensor.

Hence, we have the following linear state-space model

$$\mathbf{x}_k = \mathbf{x}_{k-1} \tag{5.9a}$$

$$\mathbf{y}_k = \mathbf{H}_{\mathcal{GP}}(\boldsymbol{\theta}_{i,k}, \boldsymbol{\theta}_t, 0)\mathbf{x}_k + \mathbf{v}_k\,, \qquad \mathbf{v}_k \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_{\mathcal{GP}}(\boldsymbol{\theta}_{i,k}, \boldsymbol{\theta}_t, 0) + \sigma_r^2 \mathbf{I}_{N_{k,m}}\right)\,, \tag{5.9b}$$

where $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{K}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_t))$ since the radius function is modeled by $\mathcal{GP}(0, k)$ with $k = k_{2\pi}, k_\pi$ or $k_{2\pi,a}$.

Note that the size of the measurement matrix and the measurement noise covariance matrix can change depending on the number of lidar range measurements available at a time point. Moreover, the matrices in the state-space model are independent of the time step $\Delta t_k = t_k - t_{k-1}$.

Algorithm 1 The Kalman filter for known pose

$\hat{\mathbf{x}}_0 \leftarrow \mathbf{0}$ ▷ Initial state estimate
$\mathbf{P}_0 \leftarrow \mathbf{K}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_t)$ ▷ Initial estimate covariance
for $k = 1, 2, \cdots$ **do**
  Get $(\mathbf{r}_k, \boldsymbol{\varphi}_k)$ ▷ Get lidar range measurements
  Get $(\mathbf{y}_k, \boldsymbol{\theta}_{i,k})$ from $(\mathbf{r}_k, \boldsymbol{\varphi}_k)$ ▷ Get radius function measurements
  $\mathbf{H}_k \leftarrow \mathbf{H}_{\mathcal{GP}}(\boldsymbol{\theta}_{i,k}, \boldsymbol{\theta}_t, 0)$ ▷ Measurement matrix
  $\mathbf{R}_k \leftarrow \mathbf{R}_{\mathcal{GP}}(\boldsymbol{\theta}_{i,k}, \boldsymbol{\theta}_t, 0) + \sigma_r^2 \mathbf{I}_{N_{k,m}}$ ▷ Measurement noise covariance
  $\boldsymbol{\nu}_k \leftarrow \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1}$ ▷ The innovation
  $\mathbf{S}_k \leftarrow \mathbf{H}_k \mathbf{P}_{k-1} \mathbf{H}_k^T + \mathbf{R}_k$ ▷ The innovation covariance
  $\mathbf{K}_k \leftarrow \mathbf{P}_{k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}$ ▷ The Kalman gain
  $\hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k \boldsymbol{\nu}_k$ ▷ The posterior state estimate
  $\mathbf{P}_k \leftarrow (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\, \mathbf{P}_{k-1}$ ▷ The posterior covariance

Since the state-space model (5.9) is linear, a Kalman filter (see Appendix B.2) is used to estimate the state $\mathbf{x} = [r(\theta_{t,n})]_{n=1}^{N_t}$. The algorithm for the resulting filter is summarized in Algorithm 1. The initial state estimate and its covariance matrix are $\hat{\mathbf{x}}_0 = \mathbf{0}$ and $\mathbf{P}_0 = \mathbf{K}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_t)$, respectively, due to the distribution $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{K}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_t))$. Since the transition matrix is the identity and there is no process noise, the predicted state estimate is equal to the previous state estimate, i.e. $\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}_{k-1}$, and $\mathbf{P}_{k|k-1} = \mathbf{P}_{k-1}$. This simplifies the algorithm for the filter, which is reduced to an iterative regression method.

The state-space model (5.9) and the associated filter (Algorithm 1) can easily be modified to the case where the states N_c, E_c and ψ_c are estimated using an external method. This approach could be used in a sensor fusion setting, and is discussed in more detail together with other possibilities in Chapter 7.

5.2 General case

We consider now the general case where the whole state of the object

$$\mathbf{x} = \begin{bmatrix} N_c & v_N & E_c & v_E & \psi_c & r_c & r(\theta_1) & r(\theta_2) & \cdots & r(\theta_{N_t}) \end{bmatrix}^T \tag{5.10}$$

is estimated based on the lidar measurements. To achieve this, a non-linear state-space is derived, where the measurement model is based on an implicit equation, and an extended Kalman filter (EKF) is used for inference (see Appendix B.3).

Two different approaches will be used to derive an explicit equation for the measurement model. The first one, which is based in the work of Wahlström and Özkan in [28], neglects the implicit dependency on the measurements of one part of the implicit equation, while the second one, which is proposed in this report, uses implicit derivation to find the measurement matrix and the measurement noise covariance matrix for the EKF.

5.2.1 Process model

The states that describe the kinematic properties of the object are

$$\mathbf{x}_c = \begin{bmatrix} N_c & v_N & E_c & v_E & \psi_c & r_c \end{bmatrix}^T , \tag{5.11}$$

and its dynamics are modeled as three decoupled one-dimensional constant velocity models (see Appendix B.1): one for the movement in the North direction, one for the movement in the East direction and one for the rotation of the object. Moreover, we assume a constant time step T. Hence, the model for $\mathbf{x}_c$ is

$$\mathbf{x}_{c,k} = \mathbf{F}_c \mathbf{x}_{c,k-1} + \mathbf{w}_{c,k} , \qquad \mathbf{w}_{c,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_c) , \tag{5.12}$$

where

$$\mathbf{F}_c = \mathbf{I}_3 \otimes \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{5.13a}$$

$$\mathbf{Q}_c = \operatorname{diag}(\sigma_N^2, \sigma_E^2, \sigma_\psi^2) \otimes \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \tag{5.13b}$$

and σ_N, σ_E and σ_ψ are the respective random noise strength for each of the decoupled constant velocity models. In particular, the course angle, which corresponds to the direction of linear velocity vector $[v_N, v_E]^T$ is independent of the heading angle ψ_c. In other words, this model takes sideslip into account.

For the distribution of the initial state, $\mathbf{x}_{c,0}$, we use the model described in Appendix B.1 for each of the three decoupled constant velocity models. Therefore $\mathbf{x}_{c,0} \sim \mathcal{N}(\hat{\mathbf{x}}_{c,0}, \mathbf{P}_{c,0})$, where

$$\hat{\mathbf{x}}_{c,0} = \begin{bmatrix} \hat{N}_{c,0} & \hat{v}_{N,0} & \hat{E}_{c,0} & \hat{v}_{E,0} & \hat{\psi}_{c,0} & \hat{r}_{c,0} \end{bmatrix}^T \tag{5.14a}$$

$$\mathbf{P}_{c,0} = \operatorname{diag}\left(\sigma^2_{\hat{N}}, \sigma^2_{\hat{E}}, \sigma^2_{\hat{\psi}}\right) \otimes \begin{bmatrix} 1 & \frac{1}{T} \\ \frac{1}{T} & \frac{2}{T^2} \end{bmatrix} \tag{5.14b}$$

and $\sigma_{\hat{N}}$, $\sigma_{\hat{E}}$ and $\sigma_{\hat{\psi}}$ are the standard deviations for the initial estimate of N_c, E_c and ψ_c, respectively.

As discussed in 5.1, the state that describes the object's extent:

$$\mathbf{x}_t = \begin{bmatrix} r(\theta_1) & r(\theta_2) & \cdots & r(\theta_{N_t}) \end{bmatrix}^T = [r(\boldsymbol{\theta}_t)]_{t=1}^{N_t} \tag{5.15}$$

is constant in time, and is modeled as a GP of the form $\mathcal{GP}(0, k(\theta, \theta'))$. Hence,

$$\mathbf{x}_{t,k} = \mathbf{x}_{t,k-1} \tag{5.16a}$$

$$\mathbf{x}_{t,0} = \mathcal{N}(\mathbf{0}, \mathbf{K}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_t)) \tag{5.16b}$$

Therefore, the process model for the whole state $\mathbf{x} = [\mathbf{x}_c^T, \mathbf{x}_t^T]^T$ is

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k \tag{5.17a}$$

$$\mathbf{x}_0 \sim \mathcal{N}\left(\begin{bmatrix} \hat{\mathbf{x}}_{c,0}^T, \mathbf{0}^T \end{bmatrix}^T, \operatorname{diag}(\mathbf{P}_{c,0}, \mathbf{K}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_t))\right), \tag{5.17b}$$

where $\mathbf{F} = \operatorname{diag}(\mathbf{F}_c, \mathbf{I}_{N_t})$ and $\mathbf{w}_k$ is white random noise with covariance $\mathbf{Q} = \operatorname{diag}(\mathbf{Q}_c, \mathbf{0}_{N_t})$.

5.2.2 Measurement model

The implicit equation for the measurement model is based on the relation between the cartesian coordinates of the lidar measurements and their corresponding radius function values. As discussed in Section 5.1, the world coordinates of a lidar range measurement r_m for an azimuth angle φ_m is

$$\mathbf{z}_m^w = r_m \begin{bmatrix} \cos(\varphi_m) \\ \sin(\varphi_m) \end{bmatrix}, \tag{5.18}$$

and under the assumption that there is no measurement noise, these coordinates correspond to a radius function value $r_i = r(\theta_i)$ for an input angle θ_i as given by

$$\mathbf{z}_m^w = \mathbf{r}_c^w + r_i \begin{bmatrix} \cos(\theta_i + \psi_c) \\ \sin(\theta_i + \psi_c) \end{bmatrix}, \tag{5.19}$$

where $\mathbf{r}_c^w = [N_c, E_c]^T$ are the coordinates of the center in the world frame and ψ_c is the heading angle (see eq. (2.2)). This relation is illustrated in Figure 5.2. It follows from Equation (5.19) that

$$|\mathbf{z}_m^w - \mathbf{r}_c^w| = r_i = r(\theta_i) \tag{5.20a}$$
$$\angle(\mathbf{z}_m^w - \mathbf{r}_c^w) = \theta_i + \psi_c\,, \tag{5.20b}$$

which implies that

$$|\mathbf{z}_m^w - \mathbf{r}_c^w| = r\left(\angle(\mathbf{z}_m^w - \mathbf{r}_c^w) - \psi_c\right) \tag{5.21a}$$
$$\mathbf{z}_m^w - \mathbf{r}_c^w = \mathbf{e}(\mathbf{z}_m^w - \mathbf{r}_c^w) r\left(\angle(\mathbf{z}_m^w - \mathbf{r}_c^w) - \psi_c\right)\,, \tag{5.21b}$$

where $\mathbf{e}(\mathbf{x}) = \frac{x}{|x|}$, $x \in \mathbf{R}^2$, is the *unit vector function*.

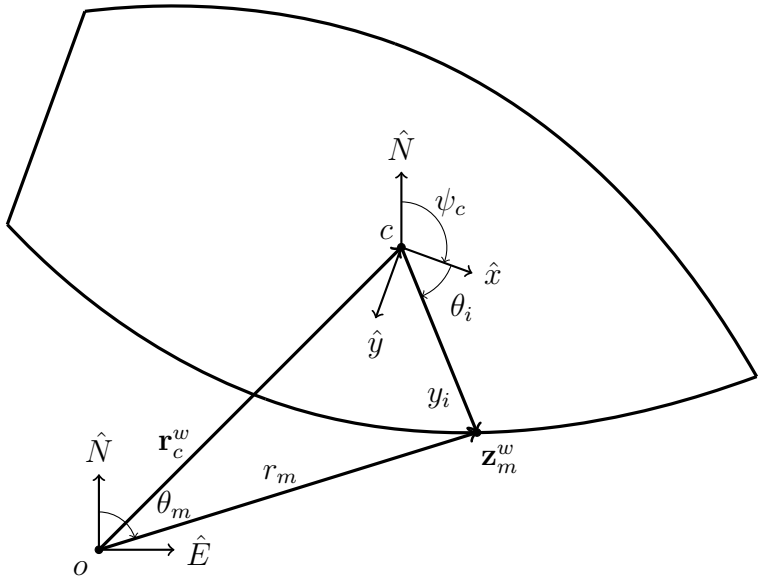


Figure 5.2: Relation between lidar measurements and radius function measurements.

Since the radius function is modeled as a GP, the identity 5.21b can be rewritten by using 4.18b as

$$\mathbf{z}_m^w = \mathbf{r}_c^w + \mathbf{e}(\mathbf{z}_m^w - \mathbf{r}_c^w)\left[\mathbf{H}_{\mathcal{GP}}(\angle(\mathbf{z}_m^w - \mathbf{r}_c^w) - \psi_c, \boldsymbol{\theta}_t, \mathbf{0})\mathbf{x}_t + \mathbf{v}_m\right]\,, \tag{5.22}$$

where $\mathbf{v}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\mathcal{GP}}(\angle(\mathbf{z}_m^w - \mathbf{r}_c^w) - \psi_c, \boldsymbol{\theta}_t, \mathbf{0}))$.

Equation (5.22) is the foundation of the measurement model. Given lidar range measurements with coordinates $\mathbf{z}_{k,1}^w, \mathbf{z}_{k,2}^w, \cdots, \mathbf{z}_{k,N_m}^w$ for a time point t_k, let the vector $\mathbf{z}_k$ be the concatenation of these coordinates, i.e.

$$\mathbf{z}_k = \begin{bmatrix} {\mathbf{z}_{k,1}^w}^T & {\mathbf{z}_{k,2}^w}^T & \cdots & {\mathbf{z}_{k,N_m}^w}^T \end{bmatrix}^T. \tag{5.23}$$

We define the functions $\mathcal{H}_{k,m}$ and $\mathcal{H}_k$ as

$$\mathcal{H}_{k,m}(\mathbf{x}, \mathbf{z}_{k,m}^w, \mathbf{v}_{k,m}) = \mathbf{r}_c^w + \mathbf{e}(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w) \left[\mathbf{H}_{\mathcal{GP}}(\angle(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w) - \psi_c, \boldsymbol{\theta}_t, \mathbf{0})\mathbf{x}_t + \mathbf{v}_{k,m}\right] \tag{5.24a}$$

$$\mathcal{H}_k(\mathbf{x}, \mathbf{z}_k, \mathbf{v}_k) = \begin{bmatrix} \mathcal{H}_{k,1}(\mathbf{x}, \mathbf{z}_{k,1}, \mathbf{v}_{k,1}) \\ \mathcal{H}_{k,2}(\mathbf{x}, \mathbf{z}_{k,2}, \mathbf{v}_{k,2}) \\ \vdots \\ \mathcal{H}_{k,N_m}(\mathbf{x}, \mathbf{z}_{k,N_m}, \mathbf{v}_{k,N_m}) \end{bmatrix}, \tag{5.24b}$$

where $\mathbf{v}_k = [\mathbf{v}_{k,1}^T, \mathbf{v}_{k,2}^T, \cdots, \mathbf{v}_{k,N_m}^T]^T$. Moreover, we define the matrix $\mathbf{R}_{\mathcal{GP},k}$ as

$$\mathbf{R}_{\mathcal{GP},k} = \operatorname{diag}\left(\mathbf{R}_{\mathcal{GP}}(\theta_{i,k,1}, \boldsymbol{\theta}_t, \mathbf{0}), \mathbf{R}_{\mathcal{GP}}(\theta_{i,k,2}, \boldsymbol{\theta}_t, \mathbf{0}), \cdots, \mathbf{R}_{\mathcal{GP}}(\theta_{i,k,N_m}, \boldsymbol{\theta}_t, \mathbf{0})\right), \tag{5.25}$$

where $\theta_{i,k,m} = \angle(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w) - \psi_c$. Note that the input angles $\theta_{i,k,m}$ and the matrix $\mathbf{R}_{\mathcal{GP},k}$ depend on the state and the measurements.

By the definition of $\mathcal{H}_{k,m}$, $\mathcal{H}_k$ and $\mathbf{R}_{\mathcal{GP},k}$, Equation (5.22) can be rewritten as

$$\mathbf{z}_k = \mathcal{H}_k(\mathbf{x}, \mathbf{z}_k, \mathbf{v}_k), \tag{5.26}$$

where $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\mathcal{GP},k})$.

Equation (5.26) is the implicit equation for the measurement model. As done in Section 5.1, we assume that the center lies inside the object. Therefore, the lidar measurements never correspond to the center, and the function $\mathcal{H}_k$ is well-defined and differentiable. Furthermore, we assume that there always exists an unique solution $\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}, \mathbf{v}_k)$ for the implicit equation 5.26, i.e.

$$\mathbf{h}_k(\mathbf{x}, \mathbf{v}_k) = \mathcal{H}_k(\mathbf{x}, \mathbf{h}_k(\mathbf{x}, \mathbf{v}_k), \mathbf{v}_k), \tag{5.27}$$

and that the function $\mathbf{h}_k$ is differentiable. In particular, we have that

$$\mathbf{z}_{k,m}^w = \mathbf{h}_{k,m}(\mathbf{x}, \mathbf{v}_{k,m}) = \mathcal{H}_{k,m}(\mathbf{x}, \mathbf{h}_{k,m}(\mathbf{x}, \mathbf{v}_{k,m}), \mathbf{v}_{k,m}), \tag{5.28}$$

where

$$\mathbf{h}_k = \begin{bmatrix} \mathbf{h}_{k,1} \\ \mathbf{h}_{k,2} \\ \vdots \\ \mathbf{h}_{k,N_m} \end{bmatrix}. \tag{5.29}$$

Although the process model (see eq. (5.17)) is linear, the solution $\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}, \mathbf{v}_k)$ of the implicit equation 5.26 is in general non-linear. Therefore, an EKF is used for inference. In particular, the Jacobian of $\mathbf{h}_k$ with respect to the state $\mathbf{x}$, which gives the measurement matrix is of interest, as well as the measurement covariance matrix.

These matrices are calculated using two different approaches. The first one, neglects that the right-hand side of 5.26 depends on $\mathbf{z}_k$. Hence, $\mathbf{h}_k = \mathcal{H}_k$, and the partial derivatives of $\mathbf{h}_k$ are calculated by differentiating $\mathcal{H}_k$. This approach is used in [28], and is called "explicit differentiation" in this report. This is in contrast to the second approach, which uses implicit differentiation to calculate the partial derivatives of $\mathbf{h}_k$.

Explicit differentiation

As discussed above, under the assumption that the dependency of the right-hand side of 5.26 on $\mathbf{z}_k$ can be neglected, it follows that $\mathbf{h}_{k,m} = \mathcal{H}_{k,m}$ and the measurement model with noise is then given by

$$\begin{aligned}
\mathbf{z}^w_{k,m} &= \mathbf{h}_{k,m}(\mathbf{x}, \mathbf{v}_{k,m}) = \mathcal{H}_{k,m}(\mathbf{x}, \mathbf{h}_{k,m}(\mathbf{x}, \mathbf{v}_{k,m}), \mathbf{v}_k) && \text{(5.30a)}\\
&= \mathbf{r}^w_c + \mathbf{e}(\mathbf{z}^w_{k,m} - \mathbf{r}^w_c)\,[\mathbf{H}(\theta_{i,k,m}, \boldsymbol{\theta}_t, \mathbf{0})\mathbf{x}_t + \mathbf{v}_{k,m}] + \mathbf{w}_{k,m} && \text{(5.30b)}\\
&= \mathbf{r}^w_c + \mathbf{e}(\mathbf{z}^w_{k,m} - \mathbf{r}^w_c)\mathbf{H}(\theta_{i,k,m}, \boldsymbol{\theta}_t, \mathbf{0})\mathbf{x}_t + \mathbf{e}(\mathbf{z}^w_{k,m} - \mathbf{r}^w_c)\mathbf{v}_{k,m} + \mathbf{w}_{k,m}\,, && \text{(5.30c)}
\end{aligned}$$

where $\mathbf{v}_{k,m} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\mathcal{GP}}(\theta_{i,k,m}, \boldsymbol{\theta}_t, \mathbf{0}))$ and $\mathbf{w}_{k,m} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{k,m})$. The noise $\mathbf{w}_{k,m}$ corresponds to the lidar measurement error given by Equation (3.5), i.e.

$$\mathbf{R}_{k,m} = \left(\mathbf{0}, \sigma_r^2 \begin{bmatrix} \cos^2(\theta_{k,m}) & \cos(\theta_{k,m})\sin(\theta_{k,m}) \\ \cos(\theta_{k,m})\sin(\theta_{k,m}) & \sin^2(\theta_{k,m}) \end{bmatrix}\right). \tag{5.31}$$

The measurement matrix at time t_k, $\mathbf{H}_{E,k}$, is obtained by the differentiation of the first two terms in by Equation (5.30c), which correspond to $\mathbf{h}_k(\mathbf{x}, \mathbf{0})$. Hence,

$$\mathbf{H}_{E,k} = \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{0}) = \begin{bmatrix} \frac{\partial \mathbf{h}_{k,1}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{0}) \\ \frac{\partial \mathbf{h}_{k,2}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{0}) \\ \vdots \\ \frac{\partial \mathbf{h}_{k,N_m}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{0}) \end{bmatrix} \tag{5.32a}$$

$$= \begin{bmatrix} \frac{\partial \mathcal{H}_{k,1}}{\partial N_c}(\mathbf{x}, \mathbf{0}) & 0 & \frac{\partial \mathcal{H}_{k,1}}{\partial E_c}(\mathbf{x}, \mathbf{0}) & 0 & \frac{\partial \mathcal{H}_{k,1}}{\partial \psi_c}(\mathbf{x}, \mathbf{0}) & 0 & \frac{\partial \mathcal{H}_{k,1}}{\partial \mathbf{x}_t}(\mathbf{x}, \mathbf{0}) \\ \frac{\partial \mathcal{H}_{k,2}}{\partial N_c}(\mathbf{x}, \mathbf{0}) & 0 & \frac{\partial \mathcal{H}_{k,2}}{\partial E_c}(\mathbf{x}, \mathbf{0}) & 0 & \frac{\partial \mathcal{H}_{k,2}}{\partial \psi_c}(\mathbf{x}, \mathbf{0}) & 0 & \frac{\partial \mathcal{H}_{k,2}}{\partial \mathbf{x}_t}(\mathbf{x}, \mathbf{0}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \mathcal{H}_{k,N_m}}{\partial N_c}(\mathbf{x}, \mathbf{0}) & 0 & \frac{\partial \mathcal{H}_{k,N_m}}{\partial E_c}(\mathbf{x}, \mathbf{0}) & 0 & \frac{\partial \mathcal{H}_{k,N_m}}{\partial \psi_c}(\mathbf{x}, \mathbf{0}) & 0 & \frac{\partial \mathcal{H}_{k,N_m}}{\partial \mathbf{x}_t}(\mathbf{x}, \mathbf{0}) \end{bmatrix}\Bigg|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}}, \tag{5.32b}$$

where $\hat{\mathbf{x}}_{k|k-1}$ is the predicted state estimate at time t_k (see Appendix B.2) and

$$\frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{r}_c^w}(\mathbf{x},\mathbf{0}) = \mathbf{I}_2 - \frac{\partial \mathbf{e}}{\partial \mathbf{x}}(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w)\mathbf{H}_{\mathcal{GP}}(\theta_{i,k,m},\boldsymbol{\theta}_t,\mathbf{0})\mathbf{x}_t - \mathbf{e}(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w)\frac{\partial \angle}{\partial \mathbf{x}}(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w)\frac{\partial \mathbf{H}_{\mathcal{GP}}}{\partial \theta}(\theta_{i,k,m},\boldsymbol{\theta}_t,\mathbf{0})\mathbf{x}_t \tag{5.33a}$$

$$\frac{\partial \mathcal{H}_{k,m}}{\partial \psi_c}(\mathbf{x},\mathbf{0}) = -\mathbf{e}(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w)\frac{\partial \mathbf{H}_{\mathcal{GP}}}{\partial \theta}(\theta_{i,k,m},\boldsymbol{\theta}_t,\mathbf{0})\mathbf{x}_t \tag{5.33b}$$

$$\frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{x}_t}(\mathbf{x},\mathbf{0}) = \mathbf{e}(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w)\mathbf{H}_{\mathcal{GP}}(\theta_{i,k,m},\boldsymbol{\theta}_t,\mathbf{0}) \tag{5.33c}$$

$$\frac{\partial \mathbf{e}}{\partial \mathbf{x}}(\mathbf{x}) = \frac{1}{|\mathbf{x}|}\mathbf{I}_2 - \frac{\mathbf{x}\mathbf{x}^T}{|\mathbf{x}|^3} \tag{5.33d}$$

$$\frac{\partial \angle}{\partial \mathbf{x}}(\mathbf{x}) = \frac{1}{|\mathbf{x}|^2}[-x_2, x_1]^T \tag{5.33e}$$

$$\frac{\partial \mathbf{H}_{\mathcal{GP}}}{\partial \theta}(\theta,\boldsymbol{\theta}_t,\mathbf{0}) = \begin{bmatrix} \frac{\partial k(\theta,\boldsymbol{\theta}_{t,1})}{\partial \theta} & \frac{\partial k(\theta,\boldsymbol{\theta}_{t,2})}{\partial \theta} & \dots \frac{\partial k(\theta,\boldsymbol{\theta}_{t,N_t})}{\partial \theta} \end{bmatrix} \mathbf{K}(\boldsymbol{\theta}_t,\boldsymbol{\theta}_t)^{-1} \tag{5.33f}$$

$$\frac{\partial k}{\partial \theta}(\theta,\theta') = \begin{cases} -\frac{\sigma_f^2}{l^2} e^{-\frac{2}{l^2}\sin^2\left(\frac{\theta-\theta'}{2}\right)} \sin(\theta-\theta')\,, & \text{if } k = k_{2\pi} \\ -\frac{\sigma_f^2}{2l^2} e^{-\frac{1}{2l^2}\sin^2(\theta-\theta')} \sin(2[\theta-\theta'])\,, & \text{if } k = k_{\pi} \\ -\frac{\sigma_f^2}{l^2} e^{-\frac{1}{2l^2}[|\operatorname{ssa}(\theta)|-|\operatorname{ssa}(\theta')|]^2} [|\operatorname{ssa}(\theta)| - |\operatorname{ssa}(\theta')|] \operatorname{sign}(\operatorname{ssa}(\theta))\,, & \text{if } k = k_{2\pi,a} \end{cases} \tag{5.33g}$$

Note that Equation (5.33a) gives the first and third columns of the measurement matrix $\mathbf{H}_k$ since $\mathbf{r}_c^w = [N_c, E_c]^T$. Furthermore, the covariance functions $k_{2\pi}$, k_π and $k_{2\pi,a}$ are not differentiable everywhere due to the random noise term, which gives a discontinuity on the line $\theta = \theta'$. Moreover, in the case of $k_{2\pi,a}$, the function $|\operatorname{ssa}(\theta)$ is not differentiable at entire multiples of π (see Figure 4.4). However, since the singular points of these covariance functions are a line in addition to a discrete set of points of the (θ,θ') plane, we consider the event of an input angle and a test angle to constitute one of these singular points as highly unlikely. Therefore, Equation (5.33g) is used despite $k_{2\pi}$, k_π and $k_{2\pi,a}$ not being differentiable everywhere.

The measurement covariance matrix is derived from the two last terms in Equation (5.30c). Hence, this matrix is

$$\mathbf{R}_{E,k} = \operatorname{diag}(\mathbf{R}_{E,k,1}, \mathbf{R}_{E,k,2}, \cdots, \mathbf{R}_{E,k,N_m})\,, \tag{5.34}$$

where

$$\mathbf{R}_{E,k,m} = \mathbf{e}(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w)\mathbf{R}_{\mathcal{GP}}(\theta_{i,k,m},\boldsymbol{\theta}_t,\mathbf{0})\mathbf{e}(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w)^T + \mathbf{R}_{k,m} \tag{5.35a}$$

$$= \begin{bmatrix}\cos(\theta_{i,k,m}+\psi_c)\\ \sin(\theta_{i,k,m}+\psi_c)\end{bmatrix} \mathbf{R}_{\mathcal{GP}}(\theta_{i,k,m},\boldsymbol{\theta}_t,\mathbf{0}) \begin{bmatrix}\cos(\theta_{i,k,m}+\psi_c)\\ \sin(\theta_{i,k,m}+\psi_c)\end{bmatrix}^T + \mathbf{R}_{k,m}. \tag{5.35b}$$

Implicit differentiation

Based on the assumption that both $\mathcal{H}_{k,m}$ and $\mathbf{h}_{k,m}$ are differentiable, the differentiation of the implicit equation $\mathbf{h}_{k,m}(\mathbf{x},\mathbf{v}_{k,m}) = \mathcal{H}_{k,m}(\mathbf{x},\mathbf{h}_{k,m}(\mathbf{x},\mathbf{v}_{k,m}),\mathbf{v}_{k,m})$ (eq. (5.28)) with respect to $\mathbf{x}$

gives

$$\frac{\partial \mathbf{h}_{k,m}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{v}_{k,m}) = \frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{h}_{k,m}(\mathbf{x}, \mathbf{v}_{k,m}), \mathbf{v}_{k,m}) + \frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{z}^w_{k,m}}(\mathbf{x}, \mathbf{h}_{k,m}(\mathbf{x}, \mathbf{v}_{k,m}), \mathbf{v}_{k,m}) \frac{\partial \mathbf{h}_{k,m}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{v}_{k,m}) \tag{5.36a}$$

$$\left[\mathbf{I}_2 - \frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{z}^w_m}(\mathbf{x}, \mathbf{z}^w_m, \mathbf{v}_{k,m})\right] \frac{\partial \mathbf{h}_{k,m}}{\partial \mathbf{x}}(\mathbf{x}) = \frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{h}_{k,m}(\mathbf{x}, \mathbf{v}_{k,m}), \mathbf{v}_{k,m}) \tag{5.36b}$$

Under the assumption that the inverse of $\mathbf{I}_2 - \frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{z}^w_m}(\mathbf{x}, \mathbf{z}^w_m, \mathbf{v}_{k,m})$ always exists, then the Jacobian of $\mathbf{h}_{k,m}$ with respect to $\mathbf{x}$ is given by

$$\frac{\partial \mathbf{h}_{k,m}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{v}_{k,m}) = \left[\mathbf{I}_2 - \frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{z}^w_m}(\mathbf{x}, \mathbf{z}^w_m, \mathbf{v}_{k,m})\right]^{-1} \frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{h}_{k,m}(\mathbf{x}, \mathbf{v}_{k,m}), \mathbf{v}_{k,m}), \tag{5.37}$$

which is the implicit derivation rule with respect to $\mathbf{x}$ for $\mathbf{z}^w_{k,m} - \mathcal{H}_{m,k}(\mathbf{x}, \mathbf{z}^w_{k,m}, \mathbf{v}_{k,m}) = \mathbf{0}$.

Comparison of the partial derivatives of $\mathcal{H}_{k,m}$ with respect to $\mathbf{z}^w_{k,m}$ and $\mathbf{r}^w_c = [N_c, E_c]^T$:

$$\begin{aligned}
\frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{z}^w_m}(\mathbf{x}, \mathbf{z}^w_m) &= \frac{\partial \mathbf{e}}{\partial \mathbf{x}}(\mathbf{z}^w_{k,m} - \mathbf{r}^w_c)\mathcal{H}(\theta_{i,k,m}, \boldsymbol{\theta}_t, \mathbf{0})\mathbf{x}_t \\
&\quad + \mathbf{e}(\mathbf{z}^w_{k,m} - \mathbf{r}^w_c)\frac{\partial \angle}{\partial \mathbf{x}}(\mathbf{z}^w_{k,m} - \mathbf{r}^w_c)\frac{\partial \mathcal{H}}{\partial \theta}(\theta_{i,k,m}, \boldsymbol{\theta}_t, \mathbf{0})\mathbf{x}_t \\
&\quad + \mathbf{e}(\mathbf{z}^w_{k,m} - \mathbf{r}^w_c)\frac{\partial \angle}{\partial \mathbf{x}}(\mathbf{z}^w_{k,m} - \mathbf{r}^w_c)\mathbf{v}_{k,m} && (5.38a) \\
\frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{r}^w_c}(\mathbf{x}, \mathbf{z}^w_m) &= \mathbf{I}_2 - \frac{\partial \mathbf{e}}{\partial \mathbf{x}}(\mathbf{z}^w_{k,m} - \mathbf{r}^w_c)\mathcal{H}(\theta_{i,k,m}, \boldsymbol{\theta}_t, \mathbf{0})\mathbf{x}_t \\
&\quad - \mathbf{e}(\mathbf{z}^w_{k,m} - \mathbf{r}^w_c)\frac{\partial \angle}{\partial \mathbf{x}}(\mathbf{z}^w_{k,m} - \mathbf{r}^w_c)\frac{\partial \mathcal{H}}{\partial \theta}(\theta_{i,k,m}, \boldsymbol{\theta}_t, \mathbf{0})\mathbf{x}_t \\
&\quad - \mathbf{e}(\mathbf{z}^w_{k,m} - \mathbf{r}^w_c)\frac{\partial \angle}{\partial \mathbf{x}}(\mathbf{z}^w_{k,m} - \mathbf{r}^w_c)\mathbf{v}_{k,m} && (5.38b) \\
&= \mathbf{I}_2 - \frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{z}^w_m}(\mathbf{x}, \mathbf{z}^w_m), && (5.38c)
\end{aligned}$$

yields that

$$\frac{\partial \mathbf{z}^w_{k,m}}{\partial \mathbf{r}^w_c} = \frac{\partial \mathbf{h}_{k,m}}{\partial \mathbf{r}^w_c}(\mathbf{x}, \mathbf{v}_{k,m}) = \mathbf{I}_2. \tag{5.39}$$

Equation (5.39) makes sense because a translation of the world coordinates of the center by $\delta \mathbf{r}^w_c$ implies an equal translation of the world coordinates of the measurements. This relation is illustrated in Figure 5.3, and can also be verified by studying Equation (5.22), which depends on the difference $\mathbf{z}^w_{k,m} - \mathbf{r}^w_c$. Note that this relation is not modelled by Equation (5.33a), which is in general different from the identity matrix.

Similarly to Equation (5.32), the measurement matrix at time t_k, $\mathbf{H}_{I,k}$, is given by the partial derivatives of $\mathbf{h}_k(\mathbf{x}, \mathbf{0})$. As a consequence of Equation (5.38) and Equation (5.39), this

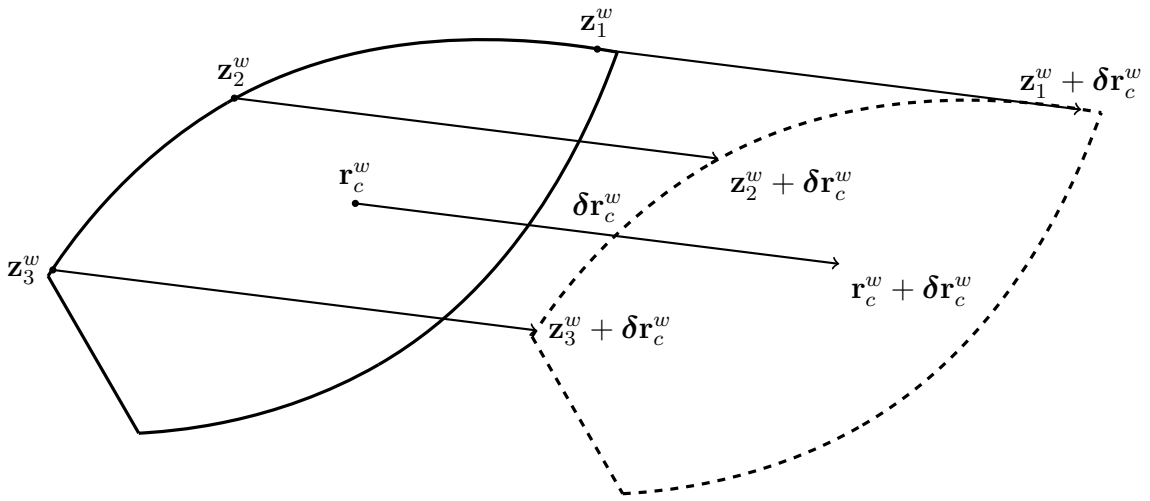


Figure 5.3: Translation of the world coordinates of the center, $\mathbf{r}_c^w$, by $\delta\mathbf{r}_c^w$, and the corresponding translation of three measurements: $\mathbf{z}_1^w$, $\mathbf{z}_2^w$ and $\mathbf{z}_3^w$.

measurement matrix is given by

$$\mathbf{H}_{I,k} = \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{0}) = \begin{bmatrix} \frac{\partial \mathbf{h}_{k,1}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{0}) \\ \frac{\partial \mathbf{h}_{k,2}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{0}) \\ \vdots \\ \frac{\partial \mathbf{h}_{k,N_m}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{0}) \end{bmatrix} \tag{5.40a}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & \frac{\partial \mathbf{h}_{k,1}}{\partial \psi_c}(\mathbf{x}, \mathbf{0}) & 0 & \frac{\partial \mathbf{h}_{k,1}}{\partial \mathbf{x}_t}(\mathbf{x}, \mathbf{0}) \\ 0 & 0 & 1 & 0 & & 0 & \\ 1 & 0 & 0 & 0 & \frac{\partial \mathbf{h}_{k,2}}{\partial \psi_c}(\mathbf{x}, \mathbf{0}) & 0 & \frac{\partial \mathbf{h}_{k,2}}{\partial \mathbf{x}_t}(\mathbf{x}, \mathbf{0}) \\ 0 & 0 & 1 & 0 & & 0 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & \frac{\partial \mathbf{h}_{k,N_m}}{\partial \psi_c}(\mathbf{x}, \mathbf{0}) & 0 & \frac{\partial \mathbf{h}_{k,N_m}}{\partial \mathbf{x}_t}(\mathbf{x}, \mathbf{0}) \\ 0 & 0 & 1 & 0 & & 0 & \end{bmatrix} \Bigg|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}}, \tag{5.40b}$$

where $\hat{\mathbf{x}}_{k|k-1}$ is the predicted state estimate at time t_k (see Appendix B.2) and

$$\frac{\partial \mathbf{h}_{k,m}}{\partial \psi}(\mathbf{x}) = -\left[\frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{r}_c^w}(\mathbf{x}, \mathbf{z}_{k,m}^w)\right]^{-1} \mathbf{e}(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w) \frac{\partial \mathbf{H}_{\mathcal{GP}}}{\partial \theta}(\theta_{i,k,m}, \boldsymbol{\theta}_t, \mathbf{0}) \mathbf{x}_t \tag{5.41a}$$

$$\frac{\partial \mathbf{h}_{k,m}}{\partial \mathbf{x}_t}(\mathbf{x}) = \left[\frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{r}_c^w}(\mathbf{x}, \mathbf{z}_{k,m}^w)\right]^{-1} \mathbf{e}(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w) \mathbf{H}_{\mathcal{GP}}(\theta_{i,k,m}, \boldsymbol{\theta}_t, \mathbf{0}). \tag{5.41b}$$

Similarly to Equation (5.37), we have that the partial derivative of $\mathbf{h}_{k,m}$ respect to the noise $\mathbf{v}_{k,m}$ is

$$\frac{\partial \mathbf{h}_{k,m}}{\partial \mathbf{v}_{\mathbf{k},\mathbf{m}}}(\mathbf{x}, \mathbf{v}_{k,m}) = \left[\mathbf{I}_2 - \frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{z}_m^w}(\mathbf{x}, \mathbf{z}_m^w, \mathbf{v}_{k,m})\right]^{-1} \frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{v}_{\mathbf{k},\mathbf{m}}}(\mathbf{x}, \mathbf{h}_{k,m}(\mathbf{x}, \mathbf{v}_{k,m}), \mathbf{v}_{k,m}) \tag{5.42a}$$

$$= \left[\frac{\partial \mathcal{H}_{k,m}}{\partial \mathbf{r}_c^w}(\mathbf{x}, \mathbf{h}_{k,m}(\mathbf{x}, \mathbf{v}_{k,m}))\right]^{-1} \mathbf{e}(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w). \tag{5.42b}$$

Therefore, the measurement covariance matrix is modeled as

$$\mathbf{R}_{I,k} = \text{diag}(\mathbf{R}_{I,k,1}, \mathbf{R}_{I,k,2}, \cdots, \mathbf{R}_{I,k,N_M}), \tag{5.43}$$

where

$$\begin{aligned}\mathbf{R}_{I,k,m} &= \left[\frac{\partial \mathbf{H_m}}{\partial \mathbf{r}_c^w}(\mathbf{x}, \mathbf{z}_m^w)\right]^{-1} \mathbf{e}(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w)\mathbf{R}_{\mathcal{GP}}(\theta_{i,k,m}, \boldsymbol{\theta}_t, \mathbf{0})\mathbf{e}(\mathbf{z}_{k,m}^w - \mathbf{r}_c^w)^T \left[\frac{\partial \mathbf{H_m}}{\partial \mathbf{r}_c^w}(\mathbf{x}, \mathbf{z}_m^w)\right]^{-T} \\ &\quad + \mathbf{R}_{k,m} \qquad (5.44a) \\ &= \left[\frac{\partial \mathbf{H_m}}{\partial \mathbf{r}_c^w}(\mathbf{x}, \mathbf{z}_m^w)\right]^{-1} \begin{bmatrix}\cos(\theta_{i,k,m} + \psi_c) \\ \sin(\theta_{i,k,m} + \psi_c)\end{bmatrix} \mathbf{R}_{\mathcal{GP}}(\theta_{i,k,m}, \boldsymbol{\theta}_t, \mathbf{0}) \begin{bmatrix}\cos(\theta_{i,k,m} + \psi_c) \\ \sin(\theta_{i,k,m} + \psi_c)\end{bmatrix}^T \left[\frac{\partial \mathbf{H_m}}{\partial \mathbf{r}_c^w}(\mathbf{x}, \mathbf{z}_m^w)\right]^{-T} \\ &\quad + \mathbf{R}_{k,m}\,, \qquad (5.44b)\end{aligned}$$

and $\mathbf{R}_{k,m}$ is the covariance matrix of the lidar measurement error given by Equation (3.5).

In other words, the covariance matrix $\mathbf{R}_{I,k,m}$ corresponds to the superposition of the effects of $\mathbf{v}_{k,m} \in \mathcal{N}(\mathbf{0}, \mathbf{R}_{\mathcal{GP}}(\theta_{i,k,m}, \boldsymbol{\theta}_t, \mathbf{0}))$ in the implicit equation (5.22) and the lidar measurement error $\mathbf{w}_{k,m} \in \mathcal{N}(\mathbf{0}, \mathbf{R}_{k,m})$.

5.2.3 Predicted measurements

An important step of the EKF is to predict measurements based on the predicted state estimate $\mathbf{x}_{k|k-1}$ in order to correct the state estimate (see Appendix B.3).

In our case, a first approach would be to use the lidar rays that correspond to the measurements to generate the predicted measurements $\hat{\mathbf{z}}_k$. This approach has two major challenges. The first one is to solve the implicit equation 5.28 for $\mathbf{z}_m^w$ for each lidar ray, which may be achieved by using root-finding algorithms since a ray is a one dimensional object. However, for a particular lidar ray and predicted state $\mathbf{x}_{k|k-1}$, this equation may or may not have a solution. The non-guaranteed existence of a predicted measurement, constitutes the second challenge: measurement association. For an arbitrary lidar ray, they might exist a measurement, but not a predicted measurement, and vice versa. Therefore it is not straightforward how these unmatched measurements or predicted measurements should be handled. Unmatched measurements could arise when the true state and the state estimate are very different. For example, during initialization of the EKF.

Due to the challenges of using lidar rays to predict measurements, and motivated by the results obtained using GP regression (see Section 4.2), the predicted measurements are generated by using the input angles defined by the measurements $\mathbf{z}_k$ and the predicted state estimate $\mathbf{x}_{k|k-1}$ together with GP regression.

Let the components of the predicted state $\mathbf{x}_{k|k-1}$ be

$$\mathbf{x}_{k|k-1} = \begin{bmatrix}\hat{N}_{c,k|k-1} & \hat{v}_{N,k|k-1} & \hat{E}_{c,k|k-1} & \hat{v}_{E,k|k-1} & \hat{\psi}_{c,k|k-1} & \hat{r}_{c,k|k-1} & \hat{\boldsymbol{x}}_{t,k|k-1}^T\end{bmatrix}^T. \tag{5.45}$$

For each measurement $\mathbf{z}_m^w$, the associated input angle is given by

$$\hat{\theta}_{i,k,m} = \angle(\mathbf{z}_m^w - \hat{\mathbf{r}}_c^w) - \hat{\psi}_{c,k|k-1}\,, \tag{5.46}$$

where $\hat{\mathbf{r}}^w_{c,k|k-1} = [\hat{N}_{c,k|k-1}, \hat{E}_{c,k|k-1}]^T$ (see Figure 5.2).

GP regression is then used to calculate the radius function value for the input angle $\hat{\theta}_{i,k,m}$ based on the predicted estimate of the object's state $\hat{\boldsymbol{x}}^T_{t,k|k-1}$ according to Equation (4.18c), which gives

$$\hat{r}_{i,k,m} = \mathbf{H}_{\mathcal{GP}}(\hat{\theta}_{i,k,m}, \theta_t, \mathbf{0})\hat{\boldsymbol{x}}_{t,k|k-1}. \tag{5.47}$$

Finally, the predicted measurement is calculated by transforming the radius function value to its corresponding coordinates in the world frame

$$\hat{\mathbf{z}}^w_m = \hat{\mathbf{r}}^w_{c,k|k-1} + \mathbf{e}(\mathbf{z}^w_m - \hat{\mathbf{r}}^w_{c,k|k-1})\hat{r}_{i,k,m} \tag{5.48a}$$

$$= \hat{\mathbf{r}}^w_{c,k|k-1} + \mathbf{e}(\mathbf{z}^w_m - \hat{\mathbf{r}}^w_{c,k|k-1})\mathbf{H}_{\mathcal{GP}}(\hat{\theta}_{i,k,m}, \theta_t, \mathbf{0})\hat{\boldsymbol{x}}_{t,k|k-1}. \tag{5.48b}$$

5.2.4 Algorithms for the extended Kalman filters

Based on the process model (see Section 5.2.1), the measurement models (see Section 5.2.2) and the method for calculating predicted measurements (see Section 5.2.3), the algorithms for the EKF using explicit and implicit differentiation are summarized in Algorithm 2 and Algorithm 3, respectively.

Algorithm 2 The EKF for the general case using explicit differenatiation

Set $\hat{\mathbf{x}}_0$ and P_0 ▷ Initialization (eq. (5.17b))
for $k = 1, 2, \cdots$ **do**
  $\hat{\mathbf{x}}_{k|k-1} \leftarrow \mathbf{F}\hat{\mathbf{x}}_{k-1}$ ▷ The predicted state estimate (eq. (5.17a))
  $\mathbf{P}_{k|k-1} \leftarrow \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}$ ▷ The predicted covariance (eq. (5.17))
  Get $\hat{\mathbf{z}}_k$ from $\mathbf{z}_k$ and $\hat{\mathbf{x}}_{k|k-1}$ ▷ The predicted measurements (eq. (5.46)-eq. (5.48))
  $\boldsymbol{\nu}_k \leftarrow \mathbf{z}_k - \hat{\mathbf{z}}_k$ ▷ The innovation
  $\mathbf{H} \leftarrow \mathbf{H}_{E,k}$ ▷ The measurement matrix (eq. (5.32)-eq. (5.33))
  $\mathbf{R} \leftarrow \mathbf{R}_{E,k}$ ▷ The measurement covariance matrix (eq. (5.34)-eq. (5.35))
  $\mathbf{S}_k \leftarrow \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}$ ▷ The innovation covariance
  $\mathbf{K}_k \leftarrow \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}_k^{-1}$ ▷ The Kalman gain
  $\hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\boldsymbol{\nu}_k$ ▷ The posterior state estimate
  $\mathbf{P}_k \leftarrow (\mathbf{I} - \mathbf{K}_k\mathbf{H})\,\mathbf{P}_{k|k-1}$ ▷ The posterior covariance

Algorithm 3 The EKF for the general case using implicit differentiation

Set $\hat{\mathbf{x}}_0$ and P_0 ▷ Initialization (eq. (5.17b))
for $k = 1, 2, \cdots$ **do**
  $\hat{\mathbf{x}}_{k|k-1} \leftarrow \mathbf{F}\hat{\mathbf{x}}_{k-1}$ ▷ The predicted state estimate (eq. (5.17a))
  $\mathbf{P}_{k|k-1} \leftarrow \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}$ ▷ The predicted covariance (eq. (5.17))
  Get $\hat{\mathbf{z}}_k$ from $\mathbf{z}_k$ and $\hat{\mathbf{x}}_{k|k-1}$ ▷ The predicted measurements (eq. (5.46)-eq. (5.48))
  $\boldsymbol{\nu}_k \leftarrow \mathbf{z}_k - \hat{\mathbf{z}}_k$ ▷ The innovation
  $\mathbf{H} \leftarrow \mathbf{H}_{I,k}$ ▷ The measurement matrix (eq. (5.33), eq. (5.40)-eq. (5.41))
  $\mathbf{R} \leftarrow \mathbf{R}_{I,k}$ ▷ The measurement covariance matrix (eq. (5.43)-eq. (5.44))
  $\mathbf{S}_k \leftarrow \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}$ ▷ The innovation covariance
  $\mathbf{K}_k \leftarrow \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}_k^{-1}$ ▷ The Kalman gain
  $\hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\boldsymbol{\nu}_k$ ▷ The posterior state estimate
  $\mathbf{P}_k \leftarrow (\mathbf{I} - \mathbf{K}_k\mathbf{H})\,\mathbf{P}_{k|k-1}$ ▷ The posterior covariance

Chapter 6

Experiments and results

Common to all experiments are the following:

- The object's extend correspond to the hull given by the "parabola" method and the parameters in Table 2.1 (see Section 2.3).
- The radius function that correspond to the object's extend is modeled as a GP with hyperpameters given by Table 4.1.
- The test angles used to discretize the radius function associated to the hull correspond to a regular discretization of the interval $[0, 2\pi)$ of 100 test angles (see Section 2.2 and see Section 4.1).
- The only lidar sensor is located at the origin of the NED frame and its parameters are summarized in Table 3.1.
- The time step is $T = 1\,\mathrm{s}$. In particular, the time step is constant
- The norms RMSE and ANEES are used to study the consistency of the filters (see Appendix B.4).

6.1 Inference using the KF for known pose: Experiment "Static ship"

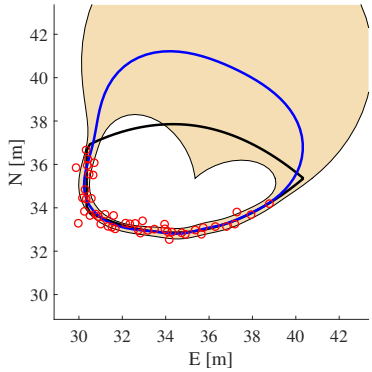
In this first experiment, we assume that the object does not move and that its position and orientation are known, and we use the KF developed in Section 5.1 to estimate the object's extend.

More precisely, the object is assumed to be located at $50\,\mathrm{m}$ to the northeast of the lidar sensor, and 8 different heading angles ranging from 0 to $315°$ are considered. For all considered heading angles, each of the GP models $\mathcal{GP}(0, k_{2\pi})$, $\mathcal{GP}(0, k_{\pi})$ and $\mathcal{GP}(0, k_{2\pi,a})$ is used to estimate the object's extend according to the KF summarized in Algorithm 1.

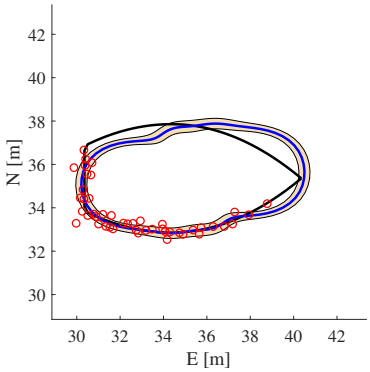
For each combination of heading angle and GP model, a Montecarlo simulation consisting of 100 simulations for $25\,\mathrm{s}$ was performed. The figures for all the results can be found in Appendix C. In Figures 8.1 to 8.3, the final estimated hull for the GP models $\mathcal{GP}(0, k_{2\pi})$, $\mathcal{GP}(0, k_{\pi})$

and $\mathcal{GP}(0, k_{2\pi,a})$ are shown. Furthermore, the RMSE and ANEES results for the Montecarlo simulations are shown in Figure 8.4 and in Figure 8.5 for each GP model, while in Figure 8.6 only the ANEES results for the GPes $\mathcal{GP}(0, k_{2\pi})$ and $\mathcal{GP}(0, k_{2\pi,a})$ are shown.

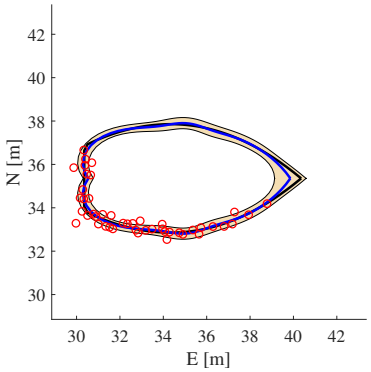
Figure 6.1 collects the results for a heading angle of $90°$, which constitutes a representative example for all the obtained results.



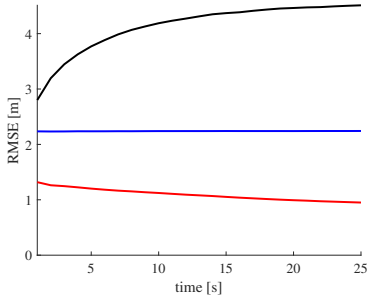
(a) Final estimated hull using GP $\mathcal{GP}(0, k_{2\pi})$: Actual hull (black), estimated hull (blue) and last set of lidar measurements (red). The shadowed region corresponds to a 99% confidence region for the estimate.



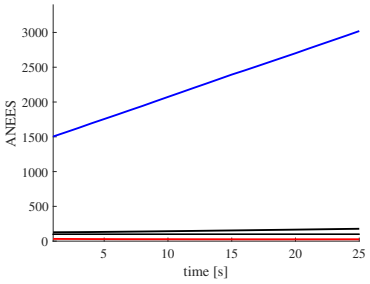
(b) Final estimated hull using GP $\mathcal{GP}(0, k_{\pi})$: Actual hull (black), estimated hull (blue) and last set of lidar measurements (red). The shadowed region corresponds to a 99% confidence region for the estimate.



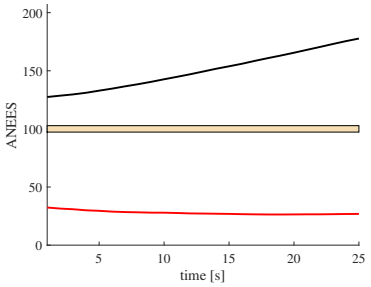
(c) Final estimated hull using GP $\mathcal{GP}(0, k_{2\pi,a})$: Actual hull (black), estimated hull(blue) and last set of lidar measurements (red). The shadowed region corresponds to a 99% confidence region for the estimate.



(d) RMSE for the GPes $\mathcal{GP}(0, k_{2\pi})$ (black), $\mathcal{GP}(0, k_{2\pi,a})$ (blue) and $\mathcal{GP}(0, k_{2\pi,a})$ (red).



(e) ANEES for the GPes $\mathcal{GP}(0, k_{2\pi})$ (black), $\mathcal{GP}(0, k_{2\pi,a})$ (blue) and $\mathcal{GP}(0, k_{\pi})$ (red). The shadowed region (strip between the black and red lines) corresponds to a 95% confidence interval for the ANEES.



(f) ANEES for the GPes $\mathcal{GP}(0, k_{2\pi})$ (black) and $\mathcal{GP}(0, k_{2\pi,a})$ (red). The shadowed region corresponds to a 95% confidence interval for the ANEES.

Figure 6.1: Results for heading angle $\psi_c = 90°$.

As illustrated in the RMSE plot (Figure 6.1d), the estimates for each GP model converge relatively fast to a final estimate. In the case of $\mathcal{GP}(0, k_{2\pi})$, this estimate is only accurate on the parts of the hull that are illuminated by the lidar sensor (Figure 6.1a) since this GP model does not assume any kind of symmetry that can improve the estimate at regions that are not scanned by the sensor. Therefore, this GP model has the largest RMSE values and the largest

confidence region for the part of the hull that is not visible for the lidar sensor. On the other hand, the estimate of $\mathcal{GP}(0, k_\pi)$ (Figure 6.1b) has a lower RMSE value and a relatively thin confidence region. However, large parts of the actual hull lie outside this confidence region. This is because the covariance function k_π forces the estimate to be symmetric about the center, while the actual hull does not posses this kind of symmetry. Another consequence of this incorrect symmetry assumption is that the values of the covariance matrix for the estimate given by the filter decrease over time. However, the estimate error remains constant. This inconsistency causes the ANEES values for $\mathcal{GP}(0, k_\pi)$ to diverge, as we can see in Figure 6.1e. In contrast to $\mathcal{GP}(0, k_\pi)$, the model given by $\mathcal{GP}(0, k_{2\pi,a})$ correctly assumes that the hull is symmetric about its longitudinal axis. Therefore the RMSE values for $\mathcal{GP}(0, k_{2\pi,a})$ are the lowest, and its hull estimate is quite accurate (Figure 6.1c)

Figure 8.6c shows the ANEES values for $\mathcal{GP}(0, k_{2\pi})$ and $\mathcal{GP}(0, k_{2\pi,a})$. Since for each simulation, the initial state vector is always the state that gives the actual hull, and the initial state estimate is always the zero vector, the simulations in the Montecarlo simulation are not independent from each other. Therefore, it cannot be expected that the ANEES values for $\mathcal{GP}(0, k_{2\pi})$ and $\mathcal{GP}(0, k_{2\pi,a})$ lie inside the confidence interval for the ANEES. Nevertheless, we can conclude that for the given set of hyperparameter values, the estimate obtained from $\mathcal{GP}(0, k_{2\pi})$ is more overconfident than the estimate obtained from $\mathcal{GP}(0, k_{2\pi,a})$.

Based on the obtained results and the discussion above, we conclude that the GP model $\mathcal{GP}(0, k_\pi)$ should not be used to estimate the hull of a maritime vessel, which is in most cases not symmetric about a point. Furthermore, the GP models $\mathcal{GP}(0, k_{2\pi})$ and $\mathcal{GP}(0, k_{2\pi,a})$ give reasonable results, and it is highly recommended to use $\mathcal{GP}(0, k_{2\pi,a})$ if the object's extend is known to be symmetric about an axis.

6.2 Inference using the EKFs

In the next two experiments, the position and orientation of the object is unknown. Therefore the EKFs developed in Section 6.1 are used to estimate the movement of the object and and extent of the object.

Common to all simulations in these experiments is that only the GP model $\mathcal{GP}(0, k_{2\pi,a})$ is used to model the object's extent beacuse of its performance in the 6.1. Furthermore, the values of the process noise strengths σ_N, σ_E and σ_ψ (see Section 5.2.1), are chosen all equal to 0.05. This is motivated by the results in [30], where such a noise strength corresponds to a vessel with little maneuverability. Moreover, the parameter values $\sigma_{\hat{N}}$, $\sigma_{\hat{E}}$ and $\sigma_{\hat{\psi}}$, which are the standard deviations for the initial estimate of N_c, E_c and ψ_c, respectively (see Section 5.2.1), are chosen as $\sigma_{\hat{N}} = \sigma_{\hat{E}} = 1\,\mathrm{m}$ and $\sigma_{\hat{\psi}} = 10°$. These parameter values are relatively low for an initial estimate if one compares them with GNSS accuracy, for example. The election of these low values is a consequence of the limited robustness of the EKFs, which will be shown and discussed later in the experiments.

6.2.1 Experiment "Random walk"

In this experiment, a "random walk" based on the process model given by Equation (5.17) is simulated. The main drawback of a direct simulation of this process model is that it may not give a realistic movement of a ship since it considers the heading angle and the linear velocity as decoupled, which is not the case in a moving ship. Therefore, in the simulations, the heading angle is taken as the direction of the linear velocity vector. In symbols, $\psi_c = \angle([v_N, v_E]^T)$.

A Montecarlo simulation consisting of 100 simulations with a duration of $60\,\mathrm{s}$ is performed to study the performance of the EKFs. In each simulation, the object starts at the NED coordinates $[20, -80]$ moving in the East direction with a velocity of $2.57\,\mathrm{m\,s^{-1}}$, which corresponds to 5 knots.

The figures for all the results can be found in Appendix D. Figure 8.7 show the trajectory of the last Montecarlo simulation. As one can see, the movement of the vessel is slowly varying. This is as expected due to the election of a low process noise strength. Furthermore, Figure 8.8 shows snapshots of the movement of the vessel and the estimation process for different time points. The performance of the EKF that uses explicit differentiation and the EKF that uses implicit differentiation are apparently similar, and their heading estimate appears to slowly diverge after a while.

This observation is supported by the results shown in Figure 8.9. Here, the estimates for the states N_c and E_c are fairly accurate. However, the estimates for the other kinematic states are not very good, specially the estimates for the heading angle.

The bad performance of the EKFs for this particular simulation is not an exception, but an example of a general issue with these estimation methods as shown by the RMSE values (Figure 8.10) and ANEES values (Figures 8.11 to 8.13) for the Montecarlo simulation. The RMSE values for the EKF with explicit differentiation start to diverge after some time, while the RMSE values for the EKF with implicit differentiation stay at a constant level, except for the heading RMSE value, which also diverges. Furthermore, the ANEES values for both methods are very high. Although, the ANEES values for the EKF with implicit differentiation are much lower that the ones for the EKF with explicit differentiation, these ANEES values are still very far away from the confidence interval for the ANEES.

6.2.2 Experiment "Turning maneuver"

In this last experiment, we consider a vessel that moves in a straight line and then makes a circular turn, while keeping a constant velocity. The motivation for this experiment is two-fold. On one side, the real trajectory of objects is usually very regular, and does not correspond to a "random walk". On the other side, we would like to study how the EKFs perform when the movement of the object does not correspond to the process model, as it is the case here since the transition from a straight line to a circle implies a step in the angular rate.

In this experiment, the vessel starts at the NED coordinates $[-40, 70]$ moving in the East direction with a velocity of $2.57\,\mathrm{m\,s^{-1}}$. After having traveled 25 meters, it turns to the starboard side and describes a circular turn with a radius of 60 meters. Finally, it continues moving west.

The trajectory of the vessel is shown in Figure 8.14.

Figure 8.15 shows snapshots of the movement of the vessel and the estimation process for different time points, and Figure 8.16 shows the estimated kinematic states for both EKFs. As one can see in these figures, the estimates of the states N_c, E_c, v_N and v_E are fairly accurate. However, the estimates of the heading angle and the yaw rate are not that good, and the heading angle estimates severely lag the actual heading angle. This causes the extent estimates to become distorted as the simulation continues, as one can see in Figure 8.15.

6.2.3 Comments on accuracy and robustness of the EKFs

As seen in the two last experiments, the accuracy of the EKFs developed in section 5.2 are limited. This is also the case for the robustness of these methods. Several simulations with velocities or process noise strengths not much higher than the values used in the presented experiments, have ended in failure due to a total divergence of the estimates of the EKFs. This has also been the case when initial state estimates that are more different than the actual state have been used.

See chapter 7 for a discussion about the poor performance of the EKFs.

Chapter 7

Conclusions and further work

Theoretical and experimental results support the use of the GP families $\mathcal{GP}(0, k_{2\pi})$, $\mathcal{GP}(0, k_{\pi})$ and $\mathcal{GP}(0, k_{2\pi,a})$ for modeling of compact and star-convex extents, where $\mathcal{GP}(0, k_{\pi})$ and $\mathcal{GP}(0, k_{2\pi,a})$ should be reserved to model extents that are symmetric about the center and extents that are symmetric about the body x-axis, respectively.

However, in the simulations, the selection of the hyperparameters values was performed heuristically and under the assumption that some knowledge about the extent is obtained during the target tracking initialization step. Further work could involve the use of well-established model selection techniques, such as leave-one-out cross-validation or log marginal likelihood maximization (see [23, ch. 5]), to relate the hyperparameters with parameters that describe the size and shape of the object's extent. By considering different extents, rule-of-thumbs could be developed that give reasonable hyperparameter values based on the information used to detect the object for the first time.

Regarding the estimation methods developed, the KF for the particular case where the pose of the object is known, performs significantly better than the EKFs for the general case. The main reasons for the poor performance of EKFs come from the measurement model used and the assumption that there is no uncertainty in the input angles when performing GP regression. In this measurement model, the covariance matrix depends indirectly from the state vector through the input angles, which gives an unusual measurement model that does not correspond to the standard state-space formulation. Furthermore, the input angles used to predict the measurements can be considerably different to the input angles for the actual measurements if the heading angle estimate is inaccurate or if the measurements are affected by strong noise. Therefore, further work could involve the development of a measurement model that takes uncertainty in the input angles into account.

Another line of work is to exploit the good performance of the KF for the particular case where the pose of the object is known, by combining it with a method that estimates the object's pose based on other measurements. This new estimation method could be just a series connection of the two abovementioned methods, or it could be a more advanced method, where the original methods are intertwined in a sensor fusion setting.

Bibliography

[1] Autoferry - ntnu. https://www.ntnu.edu/autoferry, . Accessed on 2019-05-29.

[2] Autosea - ntnu. https://www.ntnu.edu/autosea, . Accessed on 2019-05-29.

[3] Local tangent plane coordinates. https://en.wikipedia.org/wiki/Local_tangent_plane_coordinates, . Accessed on 2019-06-06.

[4] Opal™3D Lidar. http://www.neptectechnologies.com/products/opal/, . Accessed on 2019-06-04.

[5] The ReVolt DNV-GL. https://www.dnvgl.com/technology-innovation/revolt/index.html, . Accessed on 2019-05-29.

[6] Velodyne Lidar Puck™. https://velodynelidar.com/vlp-16.html, . Accessed on 2019-06-04.

[7] YARA and KONGSBERG enter into partnership. https://www.kongsberg.com/maritime/about-us/news-and-media/news-archive/2017/yara-and-kongsberg-enter-into-partnership-to-build-worlds-first-auton 2017. Accessed on 2019-05-29.

[8] YARA selects Norwegian shipbuilder VARD for zero-emission vessel Yara Birkeland. https://www.kongsberg.com/maritime/about-us/news-and-media/news-archive/2018/yara-selects-norwegian-shipbuilder-vard-for-zero-emission-vessel-yara 2018. Accessed on 2019-05-29.

[9] Marcus Baum and Uwe D Hanebeck. Extended object tracking with random hypersurface models. *IEEE Transactions on Aerospace and Electronic systems*, 50(1):149–159, 2014.

[10] Robert Grover Brown, Patrick YC Hwang, et al. *Introduction to random signals and applied Kalman filtering*, volume 3. Wiley New York, 1992.

[11] OE Drummond, SS Blackman, and KC Hell. Multiple sensor tracking of clusters and extended objects. In *Technical Proceedings*, 1988.

[12] Michael Feldmann, Dietrich Franken, and Wolfgang Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4): 1409–1420, 2011.

[13] Thor I Fossen. Marine control system-guidance, navigation and control of ships, rigs and underwater vehicles. *Marine Cybemetics*, 2002.

[14] Karl Granström and Umut Orguner. New prediction for extended targets with random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 50(2):1577–1589, 2014.

[15] Karl Granström, Christian Lundquist, and Umut Orguner. Tracking rectangular and elliptical extended targets using laser measurements. In *14th International Conference on Information Fusion*, pages 1–8. IEEE, 2011.

[16] Karl Granstrom, Marcus Baum, and Stephan Reuter. Extended object tracking: Introduction, overview and applications. *arXiv preprint arXiv:1604.00970*, 2016.

[17] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[18] Sherman Karp and Larry B Stotts. *Fundamentals of electro-optic systems design: communications, lidar, and imaging*. Cambridge University Press, 2013.

[19] Johann Wolfgang Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, 2008.

[20] Andrei N. Kolmogorov. Interpolation und Extrapolation von stationären zufälligen Folgen. *Bulletin of the Academy Sciences, USSR, Math*, 3:3–14, 1941.

[21] Danie G. Krige. A statistical approach to some mine valuations and allied problems at the witwatersrand, 1951.

[22] Hossein Mohammadi, Rodolphe Le Riche, Nicolas Durrande, Eric Touboul, and Xavier Bay. An analytic comparison of regularization methods for gaussian processes. *arXiv preprint arXiv:1602.00853*, 2016.

[23] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press: Cambridge, MA, USA, 2006. ISBN 026218253X. Accessed on 2019-05-13.

[24] Kristian Amundsen Ruud. LIDAR Extended Object Tracking of a Maritime Vessel Using an Ellipsoidal Countor Model, 2018.

[25] Kristian Amundsen Ruud, Edmund Førland Brekke, and Jo Eidsvik. Lidar extended object tracking of a maritime vessel using an ellipsoidal contour model. In *2018 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6. IEEE, 2018.

[26] Michael Schuster and Johannes Reuter. Target tracking in marine environment using automotive radar and laser range sensor. In *2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 965–970. IEEE, 2015.

[27] Michael Schuster, Johannes Reuter, and Gerd Wanielik. Probabilistic data association for tracking extended targets under clutter using random matrices. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 961–968. IEEE, 2015.

[28] Niklas Wahlström and Emre Özkan. Extended target tracking using gaussian processes. *IEEE Transactions on Signal Processing*, 63(16):4165–4178, 2015.

[29] Norbert Wiener. Extrapolation, interpolation, and smoothing of stationary time series, 1949.

[30] Erik F Wilthil, Andreas L Flåten, and Edmund F Brekke. A target tracking system for asv collision avoidance based on the pdaf. In *Sensing and Control for Autonomous Vehicles*, pages 269–288. Springer, 2017.

[31] Øystein Kaarstad Helgesen. Sensor fusion for detection and tracking of maritime vessels, 2019.

Chapter 8

Appendices

A Results about multivariate normal distributions

A.1 The conditional distribution

Let $\mathbf{x}$ and $\mathbf{y}$ be two stochastic variables that have a joint multivariate normal distribution, i.e.

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_\mathbf{x}\\ \boldsymbol{\mu}_\mathbf{x}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{\mathbf{xx}} & \boldsymbol{\Sigma}_{\mathbf{xy}}\\ \boldsymbol{\Sigma}_{\mathbf{yx}} & \boldsymbol{\Sigma}_{\mathbf{yy}},\end{bmatrix}\right) \tag{8.1}$$

where $\boldsymbol{\Sigma}_{\mathbf{yx}} = \boldsymbol{\Sigma}_{\mathbf{xy}}{}^T$. Then the conditional distribution of $\mathbf{y}$ given $\mathbf{x}$ is

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{x}}\right), \tag{8.2}$$

where

$$\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}} = \boldsymbol{\mu}_\mathbf{y} + \boldsymbol{\Sigma}_{\mathbf{yx}}\boldsymbol{\Sigma}_{\mathbf{xx}}{}^{-1}\left[\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}\right] \tag{8.3a}$$

$$\boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{x}} = \boldsymbol{\Sigma}_{\mathbf{yy}} - \boldsymbol{\Sigma}_{\mathbf{yx}}\boldsymbol{\Sigma}_{\mathbf{xx}}{}^{-1}\boldsymbol{\Sigma}_{\mathbf{xy}}. \tag{8.3b}$$

Proof: The bilinear form in the exponent of the joint probability density function $p(\mathbf{x}, \mathbf{y})$ can be decomposed as

$$\begin{bmatrix}\mathbf{x}^T - \boldsymbol{\mu}_\mathbf{x}^T & \mathbf{y}^T - \boldsymbol{\mu}_\mathbf{y}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}_{\mathbf{xx}} & \boldsymbol{\Sigma}_{\mathbf{xy}}\\ \boldsymbol{\Sigma}_{\mathbf{yx}} & \boldsymbol{\Sigma}_{\mathbf{yy}},\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}\\ \mathbf{y} - \boldsymbol{\mu}_\mathbf{y}\end{bmatrix} \tag{8.4a}$$

$$= [\mathbf{y} - \boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}}]^T\boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{x}}^{-1}[\mathbf{y} - \boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}}] + [\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}]^T\boldsymbol{\Sigma}_{\mathbf{xx}}^{-1}[\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}]. \tag{8.4b}$$

Hence, for a fixed vector $\mathbf{x}$, the conditional probability density function $p(\mathbf{y}|\mathbf{x})$ is

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x},\mathbf{y})}{\int p(\mathbf{x},\mathbf{y})\, d\mathbf{y}} \tag{8.5a}$$

$$= \frac{\exp\left(-\frac{1}{2}[\mathbf{y}-\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}}]^T\boldsymbol{\Sigma}^{-1}_{\mathbf{y}|\mathbf{x}}[\mathbf{y}-\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}}]\right)\exp\left(-\frac{1}{2}[\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}]^T\boldsymbol{\Sigma}^{-1}_{\mathbf{xx}}[\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}]\right)}{\int \exp\left(-\frac{1}{2}\mathbf{y}-\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}}]^T\boldsymbol{\Sigma}^{-1}_{\mathbf{y}|\mathbf{x}}[\mathbf{y}-\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}}]\right)\exp\left(-\frac{1}{2}[\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}]^T\boldsymbol{\Sigma}^{-1}_{\mathbf{xx}}[\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}]\right)\, d\mathbf{y}} \tag{8.5b}$$

$$= \frac{\exp\left(-\frac{1}{2}[\mathbf{y}-\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}}]^T\boldsymbol{\Sigma}^{-1}_{\mathbf{y}|\mathbf{x}}[\mathbf{y}-\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}}]\right)\exp\left(-\frac{1}{2}[\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}]^T\boldsymbol{\Sigma}^{-1}_{\mathbf{xx}}[\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}]\right)}{(2\pi)^{\frac{n}{2}}\left|\Sigma_{\mathbf{y}|\mathbf{x}}\right|^{\frac{1}{2}}\exp\left(-\frac{1}{2}[\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}]^T\boldsymbol{\Sigma}^{-1}_{\mathbf{xx}}[\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}]\right)} \tag{8.5c}$$

$$= \frac{\exp\left(-\frac{1}{2}[\mathbf{y}-\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}}]^T\boldsymbol{\Sigma}^{-1}_{\mathbf{y}|\mathbf{x}}[\mathbf{y}-\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}}]\right)}{(2\pi)^{\frac{n}{2}}\left|\Sigma_{\mathbf{y}|\mathbf{x}}\right|^{\frac{1}{2}}} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}},\boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{x}}\right). \quad \blacksquare \tag{8.5d}$$

B Discrete Kalman Filters

B.1 Discretization of the one-dimensional constant velocity model

The constant velocity model is a well-used kinematic model that models the acceleration of an object as white Gaussian random noise. Therefore, the velocity of the object is constant in average with variations that depend on the noise strength.

In the one-dimensional case, the state vector is $\mathbf{x}(t) = [x(t), v(t)]^T$, where $x(t)$ and $v(t)$ are the position and velocity of the object, respectively. Hence, the continuous state-space model is given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{G}w(t)\,, \tag{8.6}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{8.7a}$$

$$\mathbf{G} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{8.7b}$$

and $w(t)$ is white Gaussian random noise of strength σ_w, i.e.

$$w(t) \sim \mathcal{N}\left(0, \sigma_w^2 \delta(t-\tau)\right)\,, \tag{8.8}$$

where $\delta(\tau)$ is the Dirac-delta.

The solution of the differential equation in (8.6) is

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{G}w(\tau)\,d\tau. \tag{8.9}$$

If the solution is discretized with a constant time-step T, i.e. the solution is sampled at time points t_k given by $t_k = t_0 + Tk$, $k \in \mathbb{N} \cup \{0\}$, then the exact values at these time points are given by

$$\mathbf{x}(t_k) = e^{\mathbf{A}T}\mathbf{x}(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{\mathbf{A}(t_k-\tau)}\mathbf{G}w(\tau)\,d\tau. \tag{8.10}$$

By denoting $\mathbf{x}_k = \mathbf{x}(t_k)$ and

$$\mathbf{w}_k = \int_{t_{k-1}}^{t_k} e^{\mathbf{A}(t_k-\tau)}\mathbf{G}w(\tau)\,d\tau\,, \tag{8.11}$$

we can rewrite the exact discretization as

$$\mathbf{x}_k = e^{\mathbf{A}T}\mathbf{x}_{k-1} + \mathbf{w}_k. \tag{8.12}$$

Since $\mathbf{A}^n = 0$ for $n \geq 2$, it follows that

$$e^{\mathbf{A}T} = \mathbf{I} + \mathbf{A}T + \sum_{n=2}^{\infty} \frac{\mathbf{A}^n T^n}{n!} = \mathbf{I} + \mathbf{A}T = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}. \tag{8.13}$$

Moreover, $\mathbf{w}_k$ is a stochastic variables that is normally distributed with

$$\mathbb{E}[\mathbf{w}_k] = \int_{t_{k-1}}^{t_k} e^{\mathbf{A}(t_k-\tau)}\mathbf{G}\mathbb{E}[w(\tau)]\,d\tau = 0 \tag{8.14a}$$

$$\mathbb{V}[\mathbf{w}_k] = \int_{t_{k-1}}^{t_k} e^{\mathbf{A}(t_k-\tau)}\mathbf{G}\mathbb{E}[w(\tau)w(\tau)^T]\mathbf{G}^T e^{\mathbf{A}(t_k-\tau)^T}\,d\tau \tag{8.14b}$$

$$= \int_{t_{k-1}}^{t_k} \begin{bmatrix} 1 & t_k-\tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \sigma_w^2 \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_k-\tau & 1 \end{bmatrix} d\tau \tag{8.14c}$$

$$= \sigma_w^2 \int_{t_{k-1}}^{t_k} \begin{bmatrix} (t_k-\tau)^2 & t_k-\tau \\ t_k-\tau & 1 \end{bmatrix} d\tau = \sigma_w^2 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \tag{8.14d}$$

Therefore, the discrete one-dimensional constant velocity model can be stated as

$$\mathbf{x}_k = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{w}_k\,, \qquad \mathbf{w}_k \sim \mathcal{N}\left(\mathbf{0}, \sigma_w^2 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix}\right), \tag{8.15a}$$

where the initial state $\mathbf{x}_0 = \mathbf{x}(t_0)$ is assumed to be distributed as $\mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$ for some initial state estimate $\hat{\mathbf{x}}_0$ and a covariance matrix $\mathbf{P}_0$.

Initial state distribution using finite differences

Let us assume that as part of the detection process, measurements of the object position are available for the time points t_{-1} and t_0. Let us denote these estimates by $\hat{x}_0$ and $\hat{x}_{-1}$, respectively, and assume that the position measurement noise has zero mean and variance σ_x^2, and that measurements at different time points are uncorrelated.

An estimate of the velocity at t_0, $\hat{v}_0$, can be obtained by using the finite difference

$$\hat{v}_0 = \frac{\hat{x}_0 - \hat{x}_{-1}}{T} \tag{8.16}$$

If the time step T is small enough such that the finite difference is a good approximation of the velocity at t_0, it follows from this assumption and the assumptions on the measurement noise that

$$\mathbb{E}[x_0] = \hat{x}_0 \tag{8.17a}$$

$$\mathbb{E}[v_0] = \frac{\hat{x}_0 - \hat{x}_{-1}}{T} = \hat{v}_0 \tag{8.17b}$$

$$\mathbb{E}[(x_0 - \hat{x}_0)^2] = \sigma_x^2 \tag{8.17c}$$

$$\mathbb{E}[(x_0 - \hat{x}_0)(v_0 - \hat{v}_0)] = \frac{1}{T}\mathbb{E}[(x_0 - \hat{x}_0)^2] - \frac{1}{T}\mathbb{E}[(x_0 - \hat{x}_0)(x_{-1}) - \hat{x}_{-1})] = \frac{\sigma_x^2}{T} \tag{8.17d}$$

$$\mathbb{E}[(v_0 - \hat{v}_0)^2] = \frac{1}{T^2}\mathbb{E}[(x_0 - \hat{x}_0)^2] + \frac{1}{T^2}\mathbb{E}[(x_{-1} - \hat{x}_{-1})^2] = \frac{2\sigma_x^2}{T^2}. \tag{8.17e}$$

The expected values, covariances and variances of (8.17) motivate the following model for the initial state distribution

$$\mathbf{x}_0 \sim \mathcal{N}\left(\begin{bmatrix} \hat{x}_0 \\ \hat{v}_0 \end{bmatrix}, \sigma_x^2 \begin{bmatrix} 1 & \frac{1}{T} \\ \frac{1}{T} & \frac{2}{T^2} \end{bmatrix}\right). \tag{8.18}$$

B.2 The Kalman Filter (KF)

The Kalman filter is an optimal state estimator for linear systems that are subjected to white Gaussian process and measurement noise. It is optimal with respect to minimum estimation error covariance.

Consider a discrete linear state-space model of the form

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}\mathbf{u}_k + \mathbf{w}_k\,, \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{8.19a}$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k\,, \qquad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})\,, \tag{8.19b}$$

where $\mathbf{x}_k$ is the state vector of the process, $\mathbf{u}_k$ is the input vector and $\mathbf{z}_k$ is the measurement vector. The measurements and the process are subjected to white Gaussian random noise $\mathbf{v}_k$ and $\mathbf{w}_k$, respectively. Moreover, the measurement and process noise are assumed uncorrelated from the state and input vectors.

In other words, the state dynamics are given by a Markov model, and both the Markov model and the measurement likelihood are linear and Gaussian:

$$\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k \sim \mathcal{N}(\mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}\mathbf{u}_k, \mathbf{Q}) \tag{8.20a}$$

$$\mathbf{z}_k|\mathbf{x}_k \sim \mathcal{N}(\mathbf{H}\mathbf{x}_k, \mathbf{R})\,. \tag{8.20b}$$

The Kalman filter provides an estimate of the state vector for each time k, $\hat{\mathbf{x}}_k$. This is achieved by using the input and the measurements at each time k together with the linear model (8.19). One of the advantages of the Kalman filter is that its theoretical description and its practical implementation are very similar. The pseudocode for a general Kalman filter is summarized in Algorithm 4.

Algorithm 4 The Kalman filter

Set $\hat{\mathbf{x}}_0$ and P_0 ▷ Initialization
for $k = 1, 2, \cdots$ **do**
  $\hat{\mathbf{x}}_{k|k-1} \leftarrow \mathbf{F}\hat{\mathbf{x}}_{k-1} + \mathbf{G}\mathbf{u}_k$ ▷ The predicted state estimate
  $\mathbf{P}_{k|k-1} \leftarrow \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}$ ▷ The predicted covariance
  $\boldsymbol{\nu}_k \leftarrow \mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}$ ▷ The innovation
  $\mathbf{S}_k \leftarrow \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}$ ▷ The innovation covariance
  $\mathbf{K}_k \leftarrow \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}_k^{-1}$ ▷ The Kalman gain
  $\hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\boldsymbol{\nu}_k$ ▷ The posterior state estimate
  $\mathbf{P}_k \leftarrow (\mathbf{I} - \mathbf{K}_k\mathbf{H})\,\mathbf{P}_{k|k-1}$ ▷ The posterior covariance

The algorithm is initialized by setting the initial estimate for the state, $\hat{\mathbf{x}}_0$, and by setting the covariance matrix for the estimation error, $\mathbf{P}_0$. For each time k, the filter starts by making a prediction of the current state using the process model (8.19a), the previous state estimate, $\hat{\mathbf{x}}_{k-1}$, and the current input, $\mathbf{u}_k$. This gives the predicted state estimate, $\hat{\mathbf{x}}_{k|k-1}$, which is used to generate predicted measurements, $\mathbf{H}\hat{\mathbf{x}}_{k|k-1}$, as given (8.19b). The difference between the actual and the predicted measurements, $\boldsymbol{\nu}_k$, which is known as *innovation*, is used to correct the state estimate. The correction is performed by first calculating the *Kalman gain*, which is the optimal observer gain with respect to state estimation error covariance, and by adding to the predicted

state estimate the product of the Kalman gain and the innovation. This gives the posterior state estimate for time k, $\hat{\mathbf{x}}_k$. For each time k, the covariance matrices for the predicted state estimate, the innovation and the posterior state estimate are updated.

Under the following assumptions:

- The process noise and measurement noise are uncorrelated, i.e.
  $\mathbb{E}[\mathbf{v}_i^T \mathbf{w}_j] = \mathbf{0}$; (8.21a)
- The process noise and measurement noise at different times are uncorrelated, i.e.
  $\mathbb{E}[\mathbf{v}_i^T \mathbf{v}_j] = \mathbf{0}\,, \mathbb{E}[\mathbf{w}_i^T \mathbf{w}_j] = \mathbf{0}\,, i \neq j$; (8.21b)
- The system (8.19) is observable; (8.21c)
- The initial state, $\mathbf{x}_0$, is Gaussian with $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$; (8.21d)

we have that $\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_k, \mathbf{P}_k)$. In particular, the state estimate $\hat{\mathbf{x}}_k$ is unbiased. Furthermore, the estimate is optimal with respect to uncertainty in the sense that the estimate covariance matrix is minimal. In addition, the Kalman filter is globally exponentaly stable.

For the details of the Kalman filter, see the original article by Kalman [17] or the undergraduate book [10, ch. 5].

B.3 The Extended Kalman Filter (EKF)

The extended Kalman filter is a modification of the classical Kalman filter for non-linear systems, which is based on linearization about the estimated state.

Consider the non-linear state-space model

$$\mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_k\right) + \mathbf{w}_k\,, \qquad \mathbf{w}_k \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}\right) \tag{8.22a}$$

$$\mathbf{z}_k = \mathbf{h}\left(\mathbf{x}_k\right) + \mathbf{v}_k\,, \qquad \mathbf{v}_k \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}\right)\,, \tag{8.22b}$$

where $\mathbf{f}$ and $\mathbf{g}$ are continuously differentiable. This model is a generalization of the linear model (8.19). However, since the Kalman filter (Algorithm 4) depends directly on the transition matrix, $\mathbf{F}$, and the measurement matrix, $\mathbf{H}$, this state estimator cannot be applied directly to (8.22).

The extended Kalman filter uses the same scheme as the Kalman filter, where the transition and measurement matrix are generated by linearization of the non-linear model (8.22). For each time k, the transition matrix is the Jacobian of $\mathbf{f}$ at $(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)$, which corresponds to the linearization of the process model 8.22a at the previous state estimate, while the measurement matrix is the Jacobian of $\mathbf{h}$ at $\hat{\mathbf{x}}_{k|k-1}$, which corresponds to the linearization of the measurement model 8.22b at the predicted state estimate.

The pseudocode for a general extended Kalman filter is summarized in Algorithm 5.

Algorithm 5 The extended Kalman filter

Set $\hat{\mathbf{x}}_0$ and P_0 ▷ Initialization
for $k = 1, 2, \cdots$ **do**
  $\hat{\mathbf{x}}_{k|k-1} \leftarrow \mathbf{f}\left(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k\right)$ ▷ The predicted state estimate
  $\mathbf{F} \leftarrow \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)$ ▷ The prediction Jacobian
  $\mathbf{P}_{k|k-1} \leftarrow \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}$ ▷ The predicted covariance
  $\boldsymbol{\nu}_k \leftarrow \mathbf{z}_k - \mathbf{h}\left(\hat{\mathbf{x}}_{k|k-1}\right)$ ▷ The innovation
  $\mathbf{H} \leftarrow \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_{k|k-1})$ ▷ The measurement Jacobian
  $\mathbf{S}_k \leftarrow \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}$ ▷ The innovation covariance
  $\mathbf{K}_k \leftarrow \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}_k^{-1}$ ▷ The Kalman gain
  $\hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\boldsymbol{\nu}_k$ ▷ The posterior state estimate
  $\mathbf{P}_k \leftarrow \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}\right)\mathbf{P}_{k|k-1}$ ▷ The posterior covariance

Under the same assumptions as with the Kalman filter (8.21), only local exponential convergence of the state estimate to the actual state can be guaranteed. In other words, convergence can not be expected if the initial estimate is too diferent from the initial state. In addition, the extended Kalman filter is in general sub-optimal with respect to the estimate error covariance, i.e. the state estimate does not necessarily have the minimum covariance.

For the details of the extended Kalman filter, see [10, ch. 7].

B.4 Filter consistency

A systematic way of studying the performance of a KF or EKF is to measure the error between the actual state and the estimated state by means of a norm. Due to the stochastic nature of the state-space formulations of the KF and EKF, one gets a better picture of the filter performance if this norm takes the average over the results obtained from a Montecarlo simulation, which is a large set of independent simulations of the process.

One of the most well-used norms is the *Root-Mean-Square-Error* (RMSE), which is given by

$$\text{RMSE}_{j,k} = \sqrt{\frac{1}{N_{\text{sim}}} \sum_{n=1}^{N_{\text{sim}}} (\hat{\mathbf{x}}_{n,j,k} - \mathbf{x}_{n,j,k})^2}\,, \tag{8.23}$$

where N_{sim} is the number of simulations of the Montecarlo simulation, n corresponds to a particular simulation, k corresponds to a time point and j to a particular state. Therefore, for a fixed j, $\text{RMSE}_{j,k}$ gives how the RMSE for the particular state develops over time.

Another well-used norm is the *average normalized estimation error squared* (ANEES), which is defined as

$$\text{ANEES}_k = \frac{1}{N_{\text{sim}}} \sum_{n=1}^{N_{\text{sim}}} (\hat{\mathbf{x}}_{n,k} - \mathbf{x}_{n,k})^T \mathbf{P}_{n,k}^{-1} (\hat{\mathbf{x}}_{n,k} - \mathbf{x}_{n,k})\,, \tag{8.24}$$

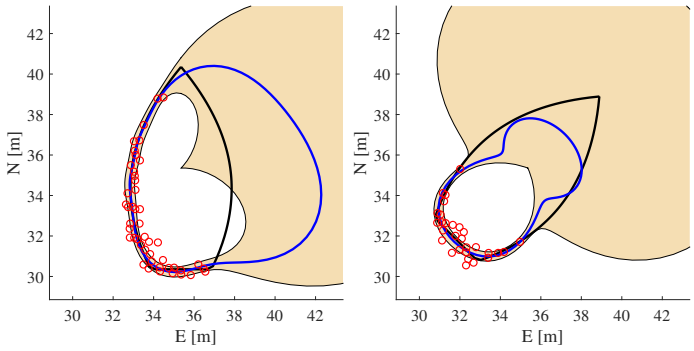
which can be reduced to a particular state by only taking the corresponding state component and diagonal entry in the covariance matrix, as given by

$$\text{ANEES}_{j,k} = \frac{1}{N_{\text{sim}}} \sum_{n=1}^{N_{\text{sim}}} (\hat{\mathbf{x}}_{n,j,k} - \mathbf{x}_{n,j,k})^T \mathbf{P}_{n,k}[j,j]^{-1} (\hat{\mathbf{x}}_{n,j,k} - \mathbf{x}_{n,j,k}). \tag{8.25}$$

Under the Kalman filter assumption (8.21), we have that $\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_k, \mathbf{P}_k)$. Hence, ANEES is a Gamma distribution with scale parameter $\frac{2}{N_{\text{sim}}}$ and shape parameter $\frac{N_{\text{sim}} N}{2}$, where N is the number of states considered.

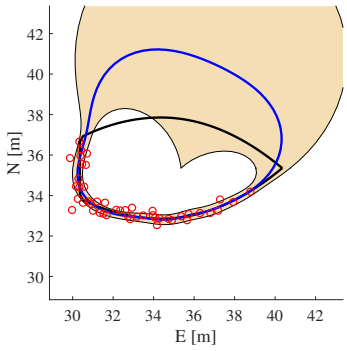
Because of this distribution, the experimental values obtained for ANEES are compared to a confidence interval for its distribution. If the experimental values lie inside these confidence interval, one has then show that the filter is consistent. If the values lie above the confidence interval, the estimate is overconfident, and if the values lie below the confidence interval, the estimate is underconfident.

C Results for experiment ”Static ship”

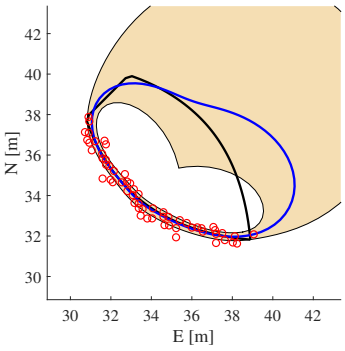


(a) $\psi_c = 0°$.

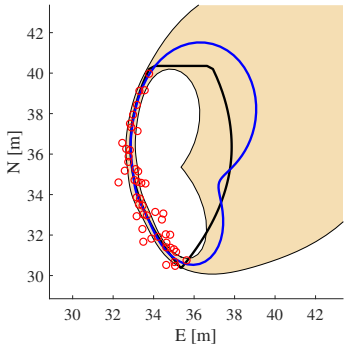
(b) $\psi_c = 45°$.



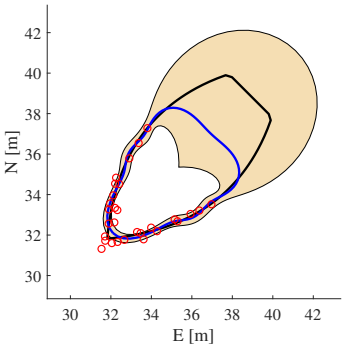
(c) $\psi_c = 90°$.



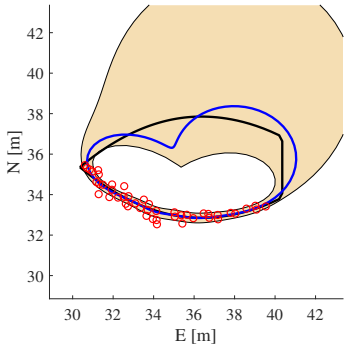
(d) $\psi_c = 135°$.



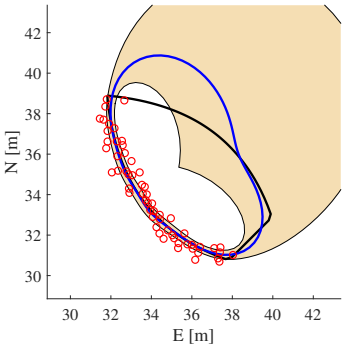
(e) $\psi_c = 180°$.



(f) $\psi_c = 225°$.

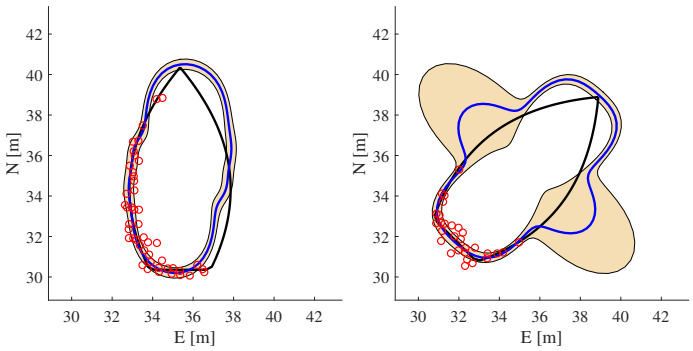


(g) $\psi_c = 270°$.



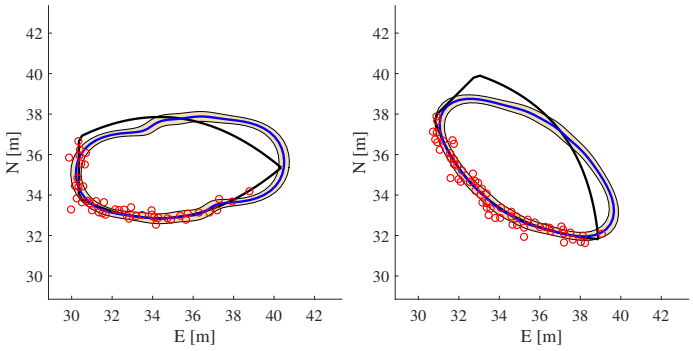
(h) $\psi_c = 315°$.

Figure 8.1: Final estimated hull using GP $\mathcal{GP}(0, k_{2\pi})$ after 25 iterations: Actual hull (black), estimated hull (blue) and last set of lidar measurements (red) are shown for different heading angles ψ_c. The shadowed region corresponds to a 99% confidence region for the estimate.



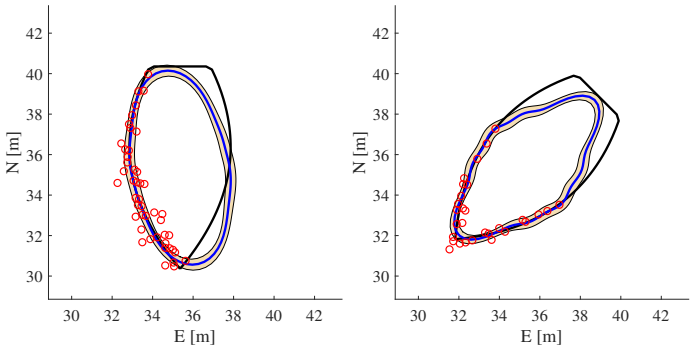
(a) $\psi_c = 0°$.

(b) $\psi_c = 45°$.



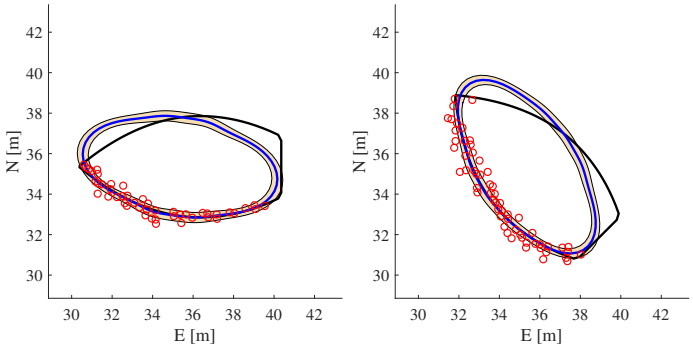
(c) $\psi_c = 90°$.

(d) $\psi_c = 135°$.



(e) $\psi_c = 180°$.

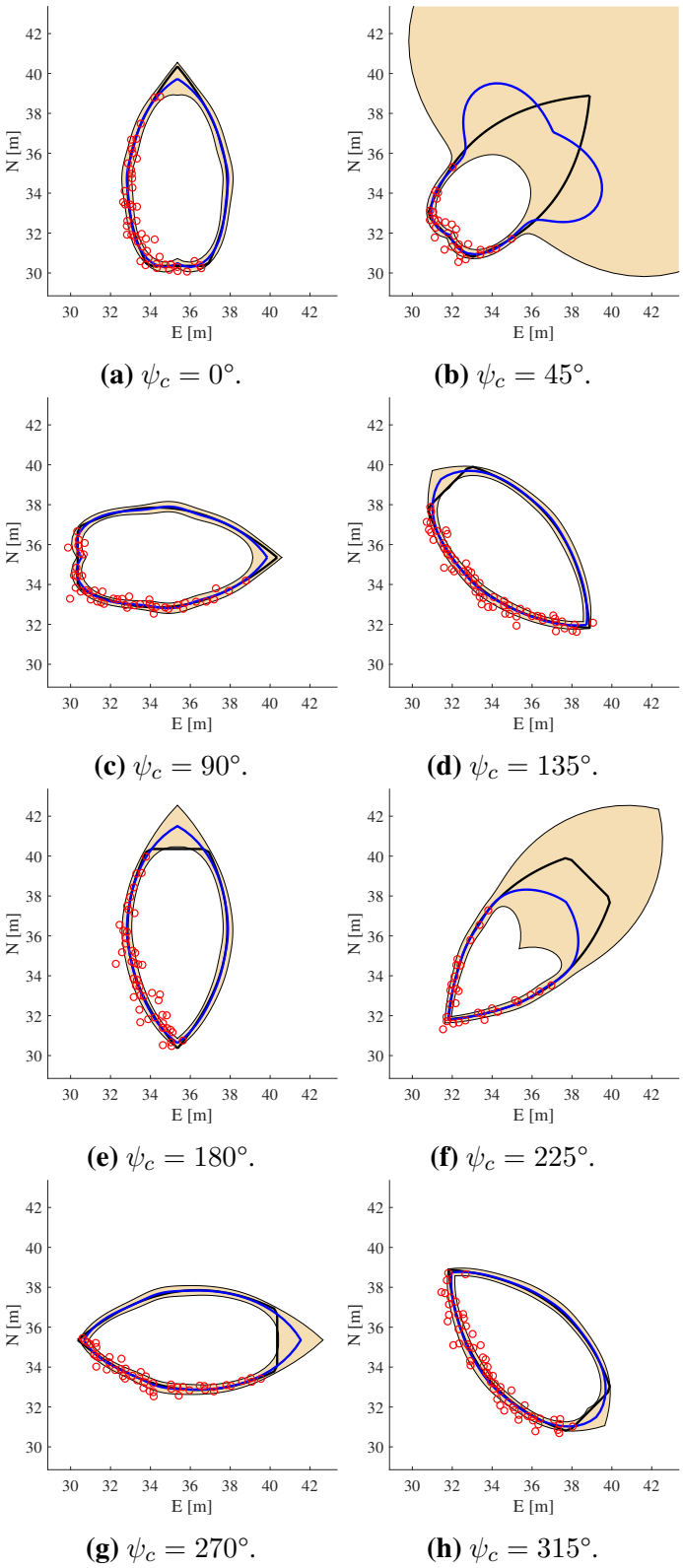
(f) $\psi_c = 225°$.



(g) $\psi_c = 270°$.

(h) $\psi_c = 315°$.

Figure 8.2: Final estimated hull using GP $\mathcal{GP}(0, k_\pi)$ after 25 iterations: Actual hull (black), estimated hull (blue) and last set of lidar measurements (red) are shown for different heading angles ψ_c. The shadowed region corresponds to a 99% confidence region for the estimate.



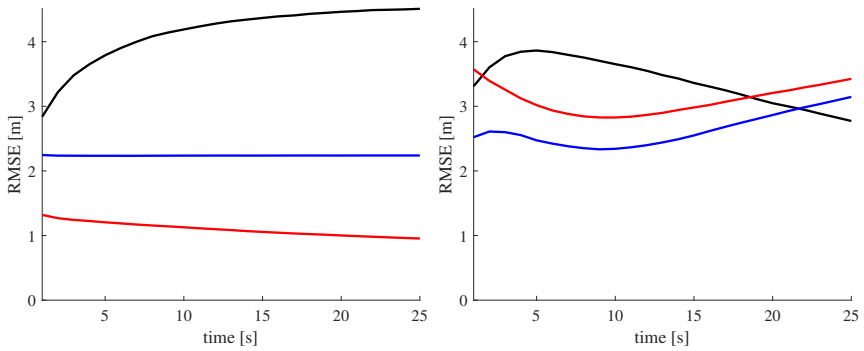
(a) $\psi_c = 0°$. **(b)** $\psi_c = 45°$.

(c) $\psi_c = 90°$. **(d)** $\psi_c = 135°$.

(e) $\psi_c = 180°$. **(f)** $\psi_c = 225°$.

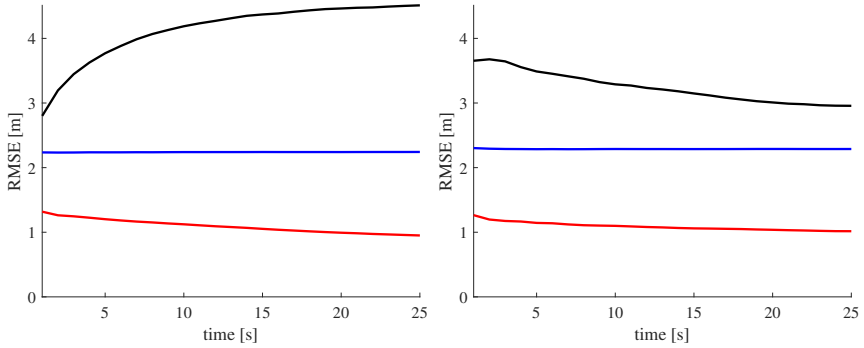
(g) $\psi_c = 270°$. **(h)** $\psi_c = 315°$.

Figure 8.3: Final estimated hull using GP $\mathcal{GP}(0, k_{2\pi,a})$ after 25 iterations: Actual hull (black), estimated hull (blue) and last set of lidar measurements (red) are shown for different heading angles ψ_c. The shadowed region corresponds to a 99% confidence region for the estimate.



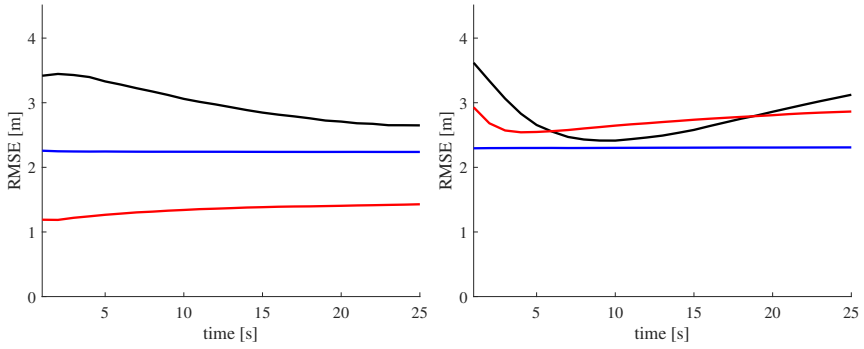
(a) $\psi_c = 0°$.

(b) $\psi_c = 45°$.



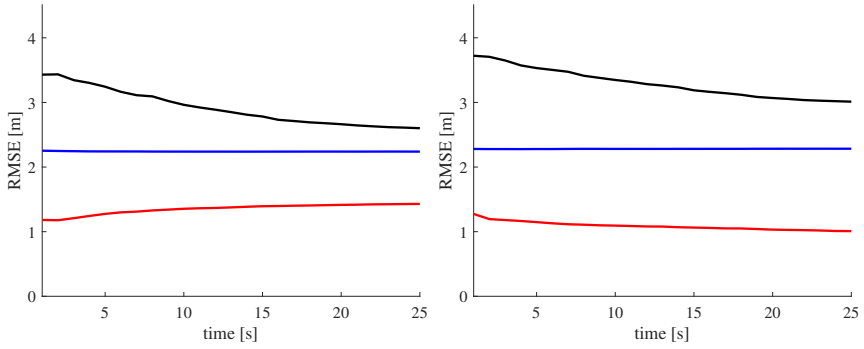
(c) $\psi_c = 90°$.

(d) $\psi_c = 135°$.



(e) $\psi_c = 180°$.

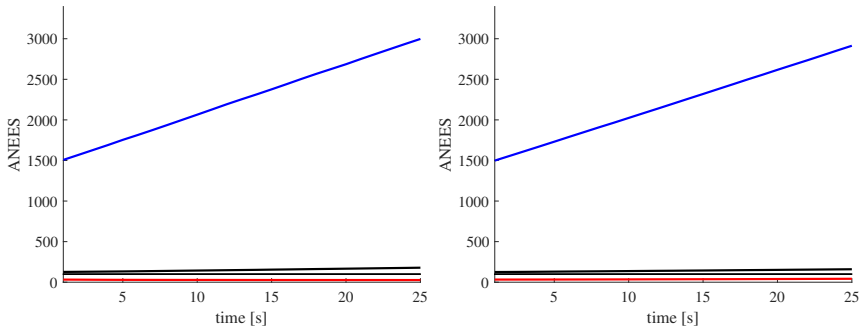
(f) $\psi_c = 225°$.



(g) $\psi_c = 270°$.

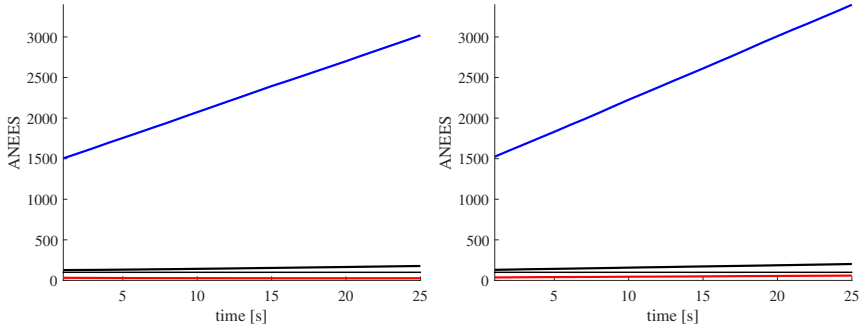
(h) $\psi_c = 315°$.

Figure 8.4: RMSE for the GPes $\mathcal{GP}(0, k_{2\pi})$ (black), $\mathcal{GP}(0, k_{\pi,a})$ (blue) and $\mathcal{GP}(0, k_{2\pi,a})$ (red). Results for 100 simulations.



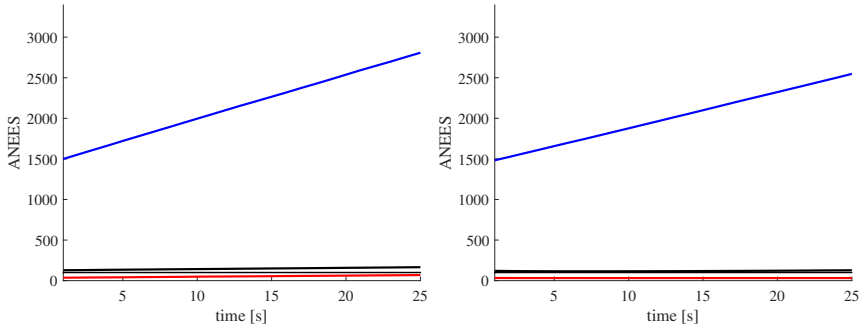
(a) $\psi_c = 0°$.

(b) $\psi_c = 45°$.



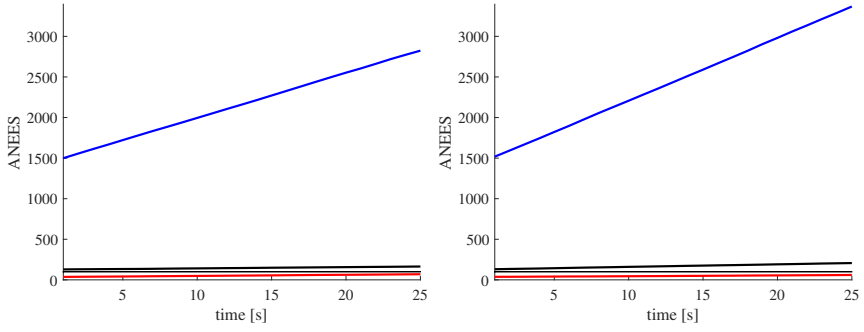
(c) $\psi_c = 90°$.

(d) $\psi_c = 135°$.



(e) $\psi_c = 180°$.

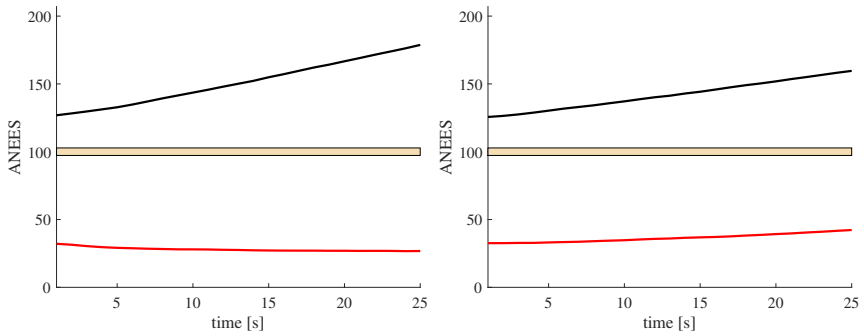
(f) $\psi_c = 225°$.



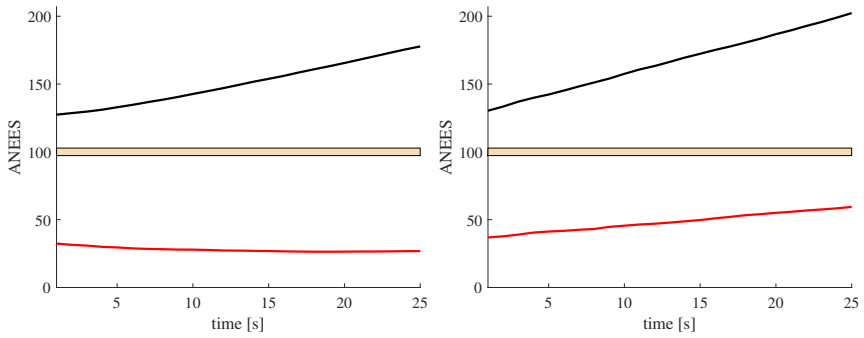
(g) $\psi_c = 270°$.

(h) $\psi_c = 315°$.

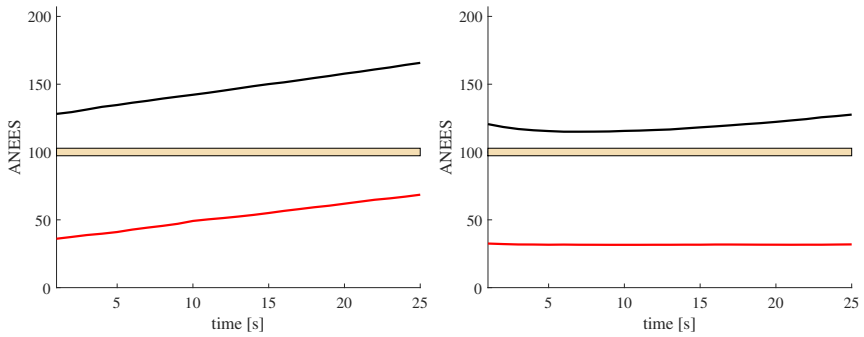
Figure 8.5: ANEES for the GPes $\mathcal{GP}(0, k_{2\pi})$ (black), $\mathcal{GP}(0, k_{\pi,a})$ (blue) and $\mathcal{GP}(0, k_{2\pi,a})$ (red). The shadowed region (strip between the black and red lines) corresponds to a 95% confidence interval for the ANEES. Results for 100 simulations.



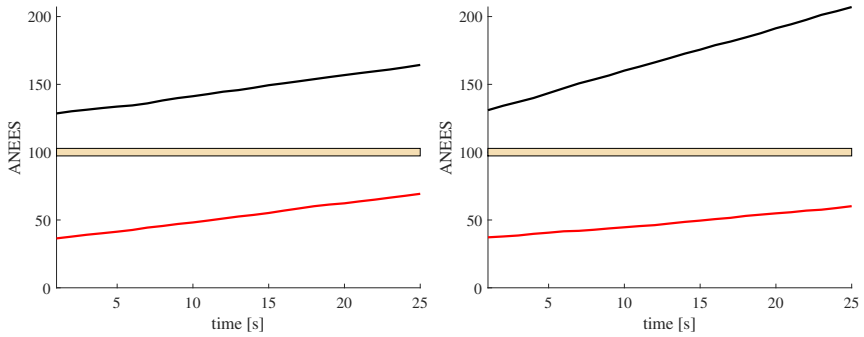
(a) $\psi_c = 0°$. **(b)** $\psi_c = 45°$.



(c) $\psi_c = 90°$. **(d)** $\psi_c = 135°$.



(e) $\psi_c = 180°$. **(f)** $\psi_c = 225°$.



(g) $\psi_c = 270°$. **(h)** $\psi_c = 315°$.

Figure 8.6: ANEES for the GPes $\mathcal{GP}(0, k_{2\pi})$ (black) and $\mathcal{GP}(0, k_{2\pi,a})$ (red). The shadowed region corresponds to a 95% confidence interval for the ANEES. Results for 100 simulations.

D Results for experiment ”Random walk”

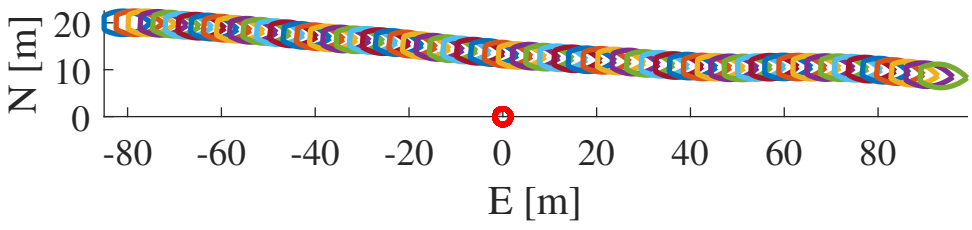
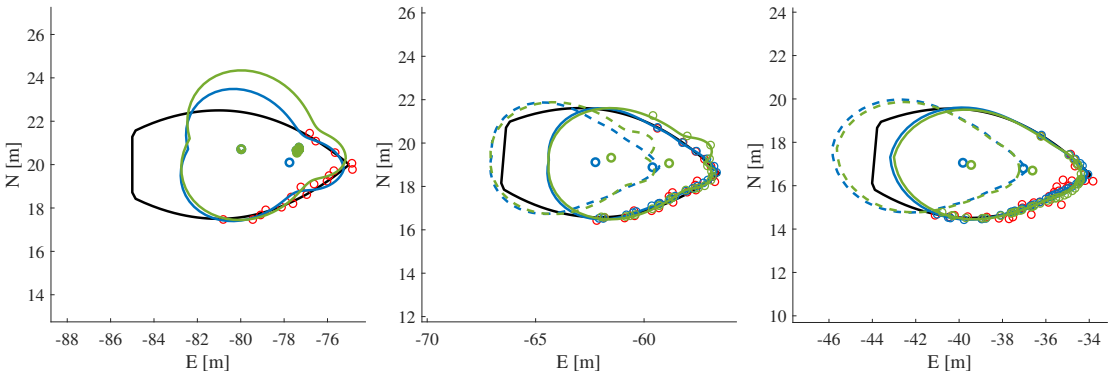


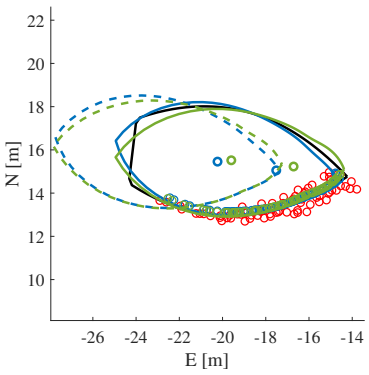
Figure 8.7: Trajectory of a random walk: The position of the hull for each time point (different colors) and the position of lidar sensor (red circle).



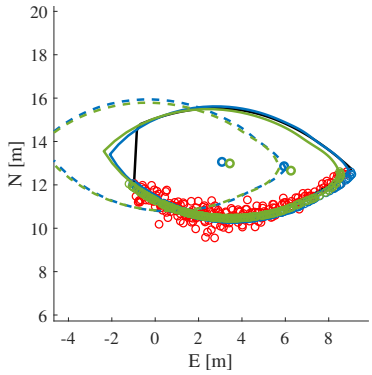
(a) $t_k = 1\,\mathrm{s}$.

(b) $t_k = 8\,\mathrm{s}$.

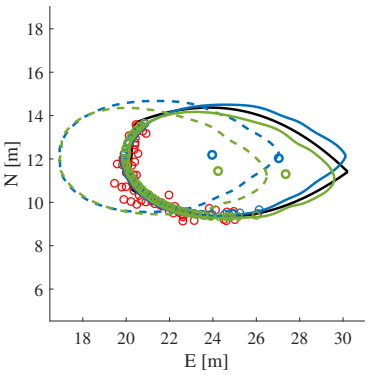
(c) $t_k = 16\,\mathrm{s}$.



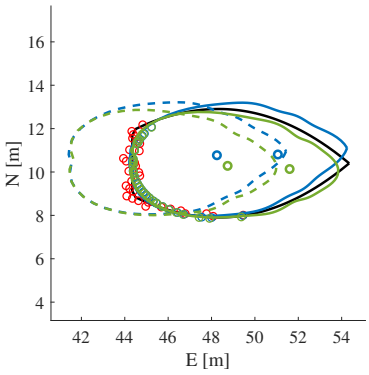
(d) $t_k = 23\,\mathrm{s}$.



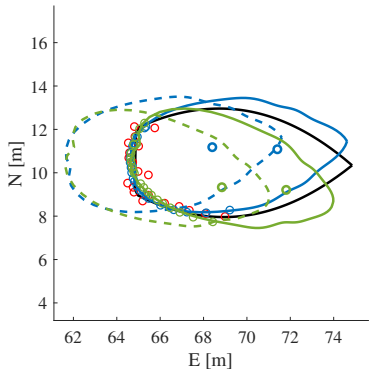
(e) $t_k = 31\,\mathrm{s}$.



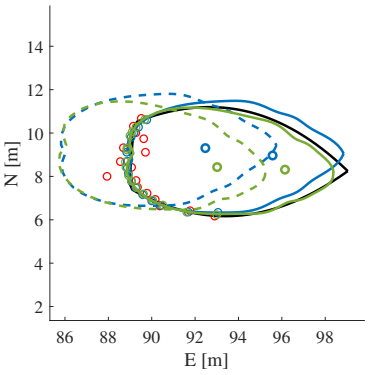
(f) $t_k = 38\,\mathrm{s}$.



(g) $t_k = 46\,\mathrm{s}$.

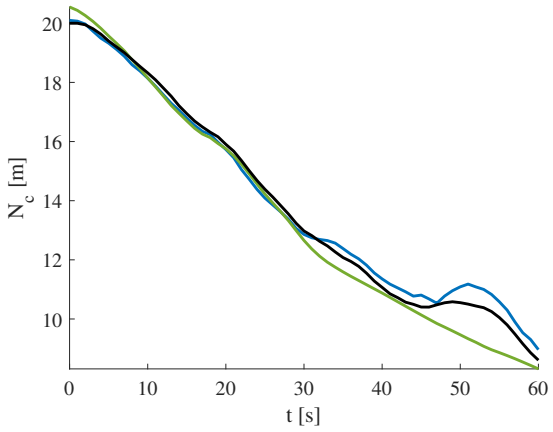


(h) $t_k = 53\,\mathrm{s}$.

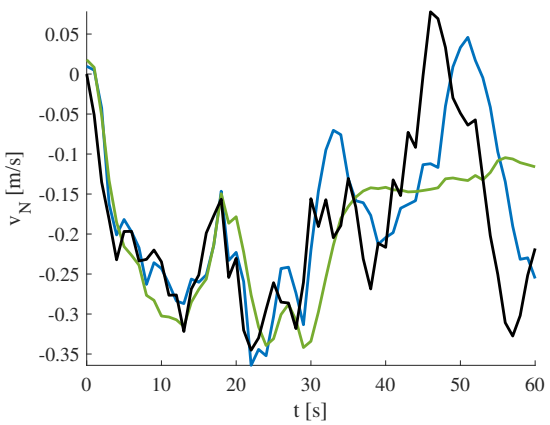


(i) $t_k = 60\,\mathrm{s}$.

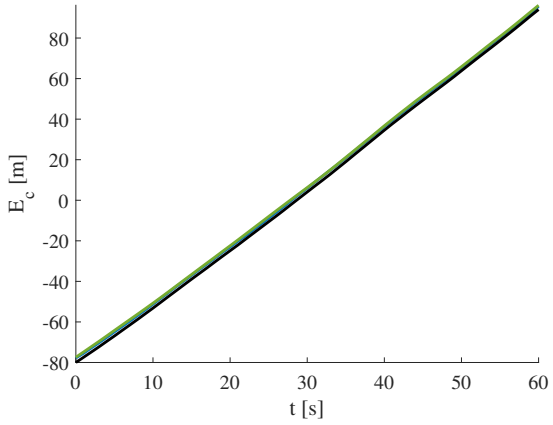
Figure 8.8: Snapshots for different time points of the lidar measurements (red), the true hull position (black) and the estimated hull position using the measurement model with explicit differentiation (blue) and with implicit differentiation (green). The solid blue and green lines correspond to the current estimate, while the dashed lines correspond to the estimate for the previous time point. The estimated positions of the center are shown as isolated blue and green circles, while the blue and green circles correspond to predict measurements.



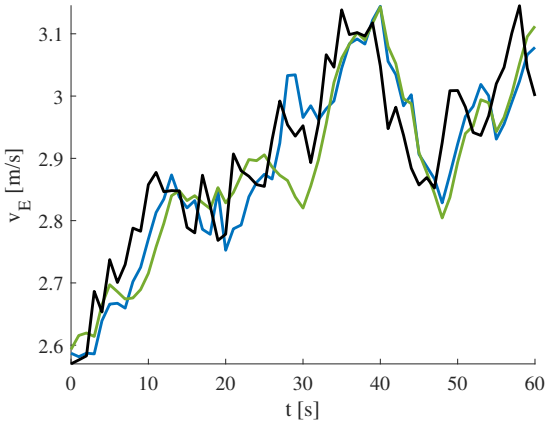
(a) North coordinate of the center, N_c.



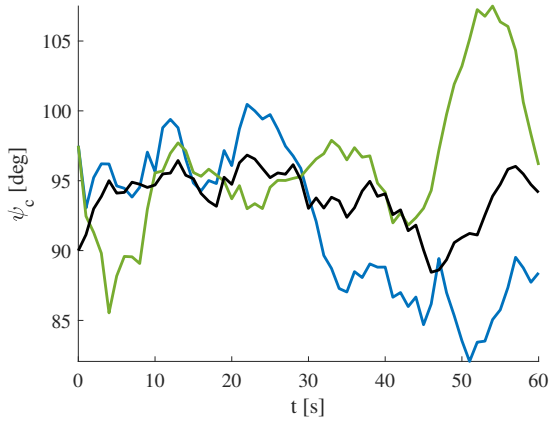
(b) Velocity in the North direction, v_N



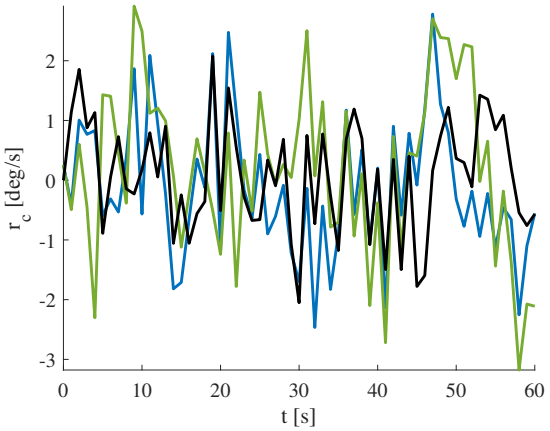
(c) East coordinate of the center E_c.



(d) Velocity in the East direction, v_E

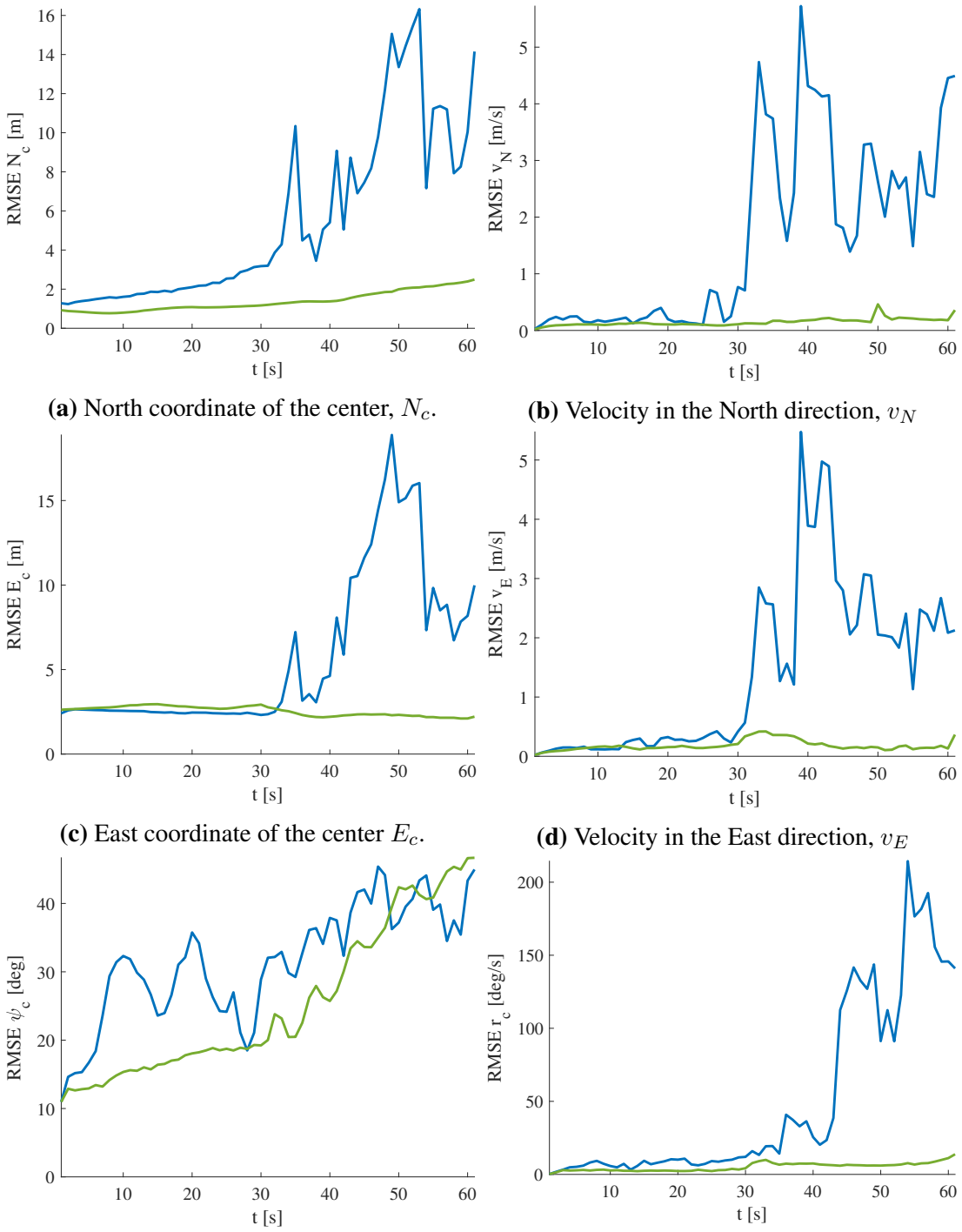


(e) Heading angle, ψ_c.



(f) Angular rate, r_c.

Figure 8.9: True state values (black) and estimated values using the measurement model with explicit differentiation (blue) and with implicit differentiation (green).



(a) North coordinate of the center, N_c.

(b) Velocity in the North direction, v_N

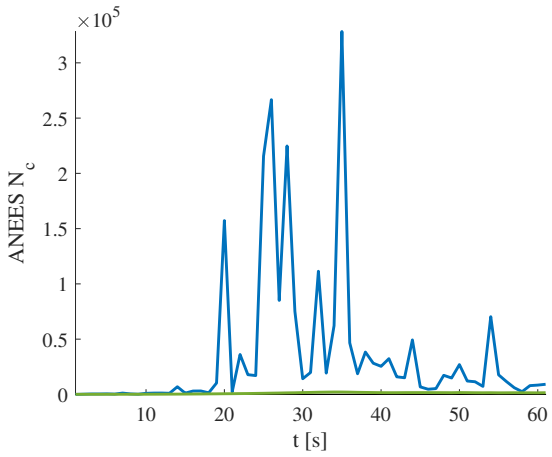
(c) East coordinate of the center E_c.

(d) Velocity in the East direction, v_E

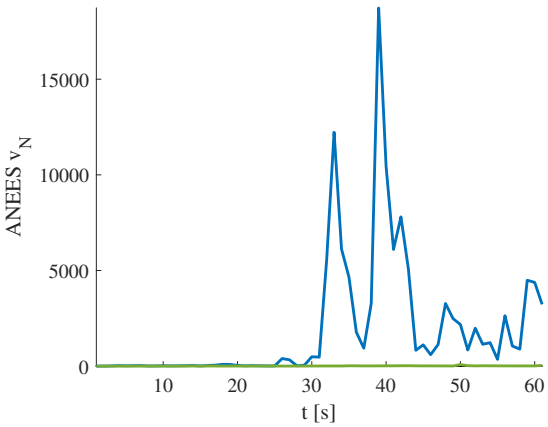
(e) Heading angle, ψ_c.

(f) Angular rate, r_c.

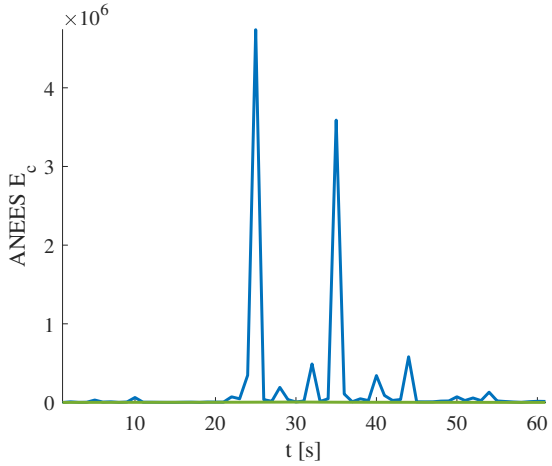
Figure 8.10: RMSE of different states: For the measurement model with explicit differentiation (blue) and with implicit differentiation (green).



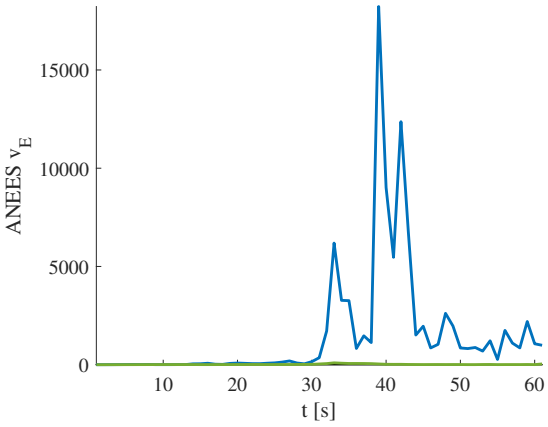
(a) North coordinate of the center, N_c.



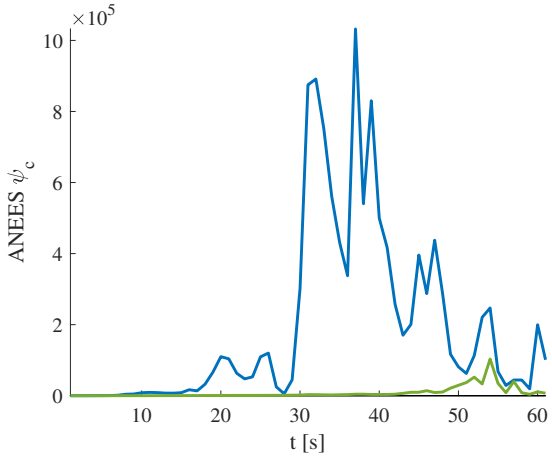
(b) Velocity in the North direction, v_N



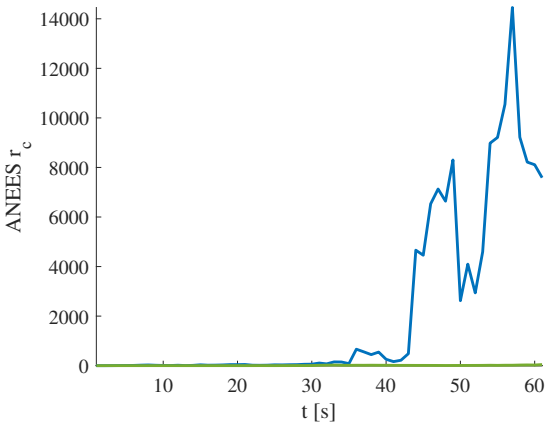
(c) East coordinate of the center E_c.



(d) Velocity in the East direction, v_E

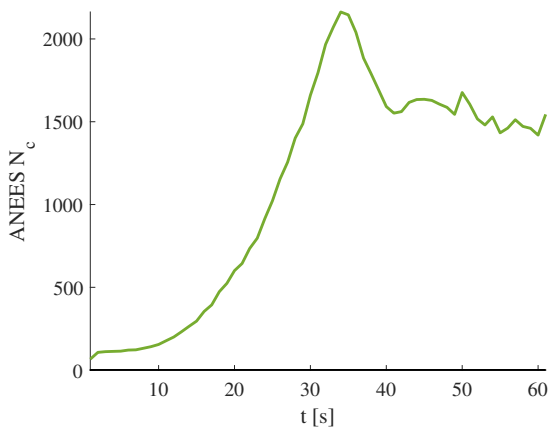


(e) Heading angle, ψ_c.

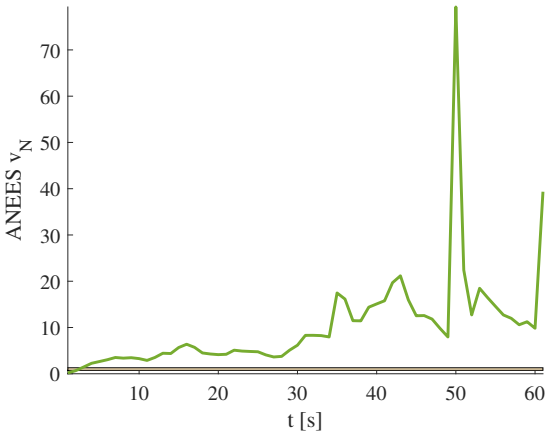


(f) Angular rate, r_c.

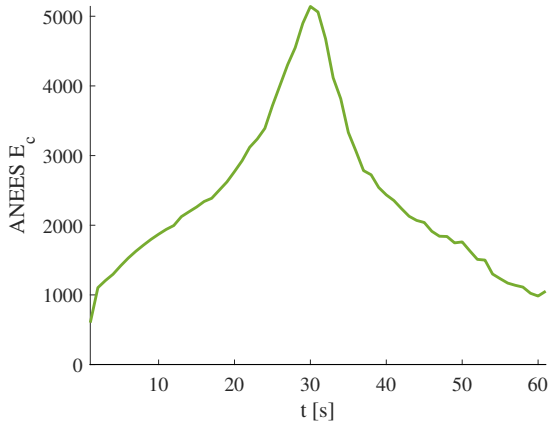
Figure 8.11: ANEES of different states: For the measurement model with explicit differentiation (blue) and with implicit differentiation (green).



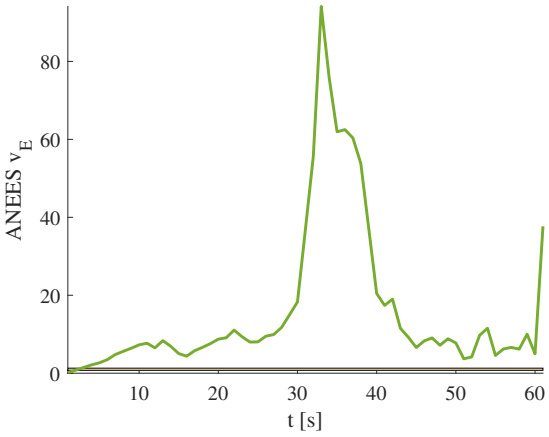
(a) North coordinate of the center, N_c.



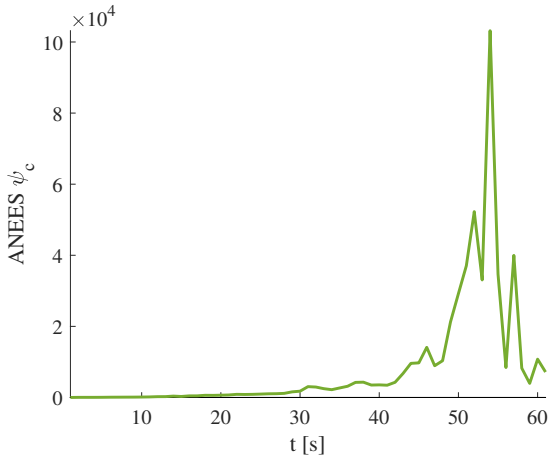
(b) Velocity in the North direction, v_N



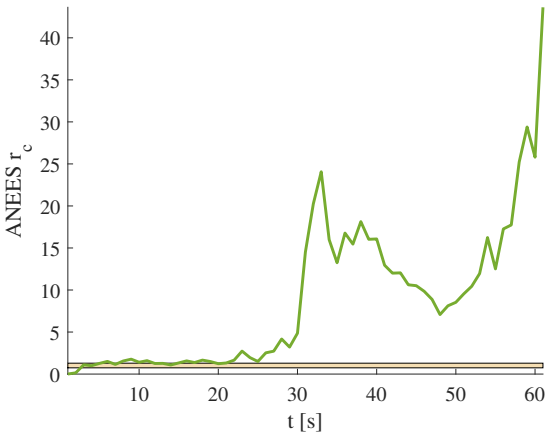
(c) East coordinate of the center E_c.



(d) Velocity in the East direction, v_E

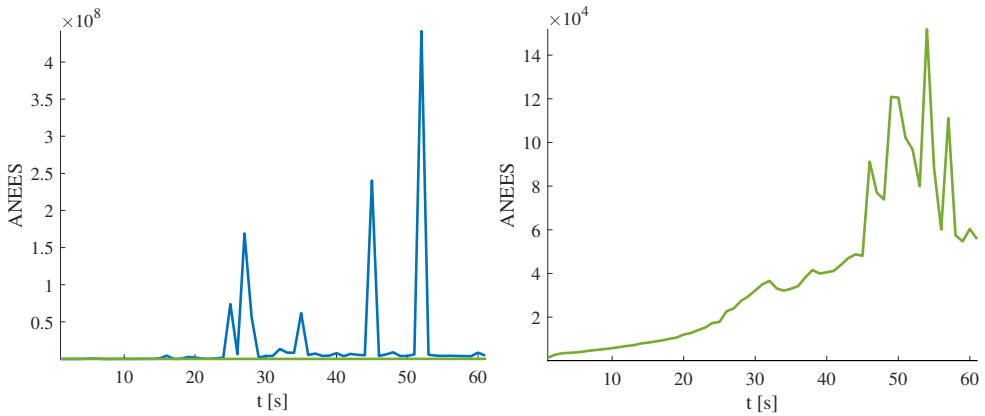


(e) Heading angle, ψ_c.



(f) Angular rate, r_c.

Figure 8.12: ANEES of different states only for the measurement model with implicit differentiation (green). A 95% confidence region for the ANEES is also shown, but is only visible for v_N, V_E and r_c.



(a) For both methods. **(b)** Only for implicit differentiation.

Figure 8.13: ANEES of the whole state vector $\mathbf{x}_c$: For the measurement model with explicit differentiation (blue) and with implicit differentiation (green).

E Results for experiment ”Turning ship”

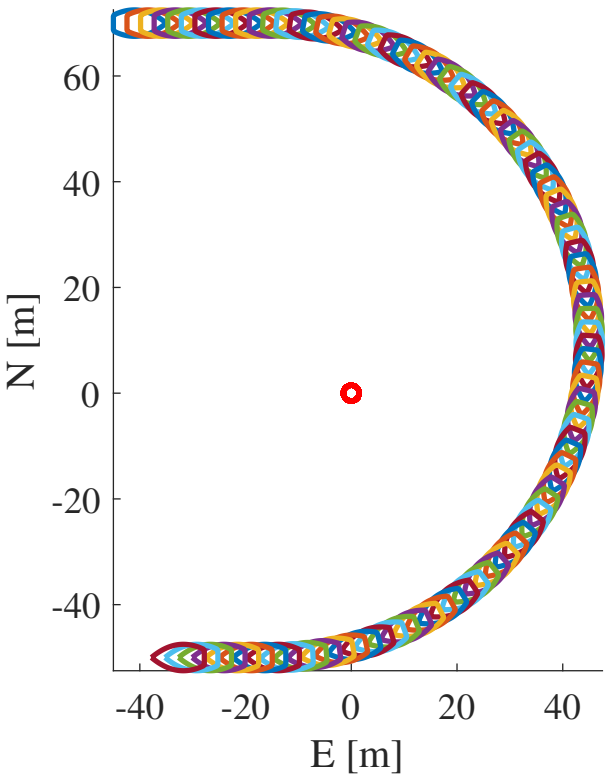
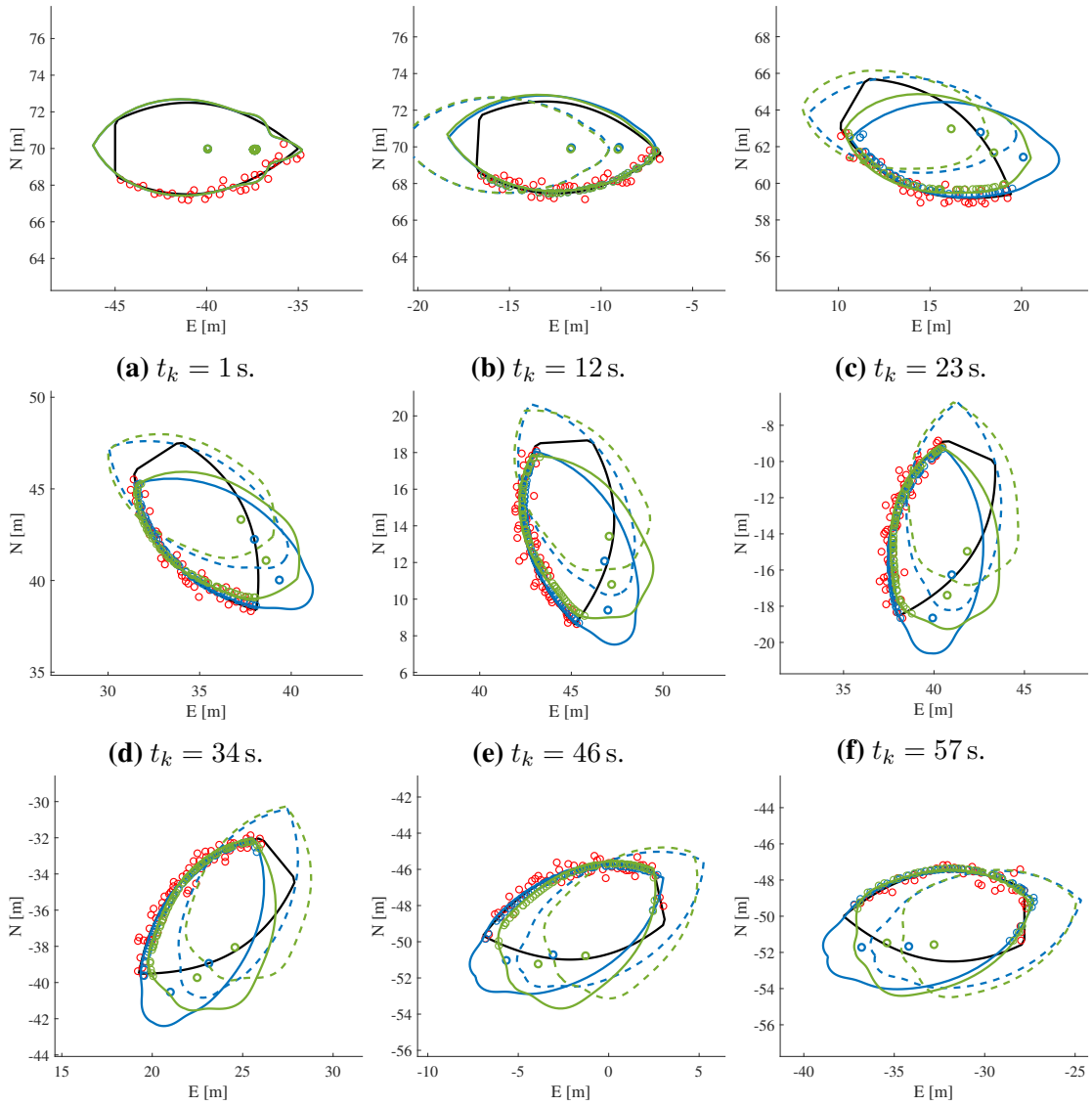


Figure 8.14: Trajectory of the turn: The position of the hull for each time point (different colors) and the position of lidar sensor (red circle).

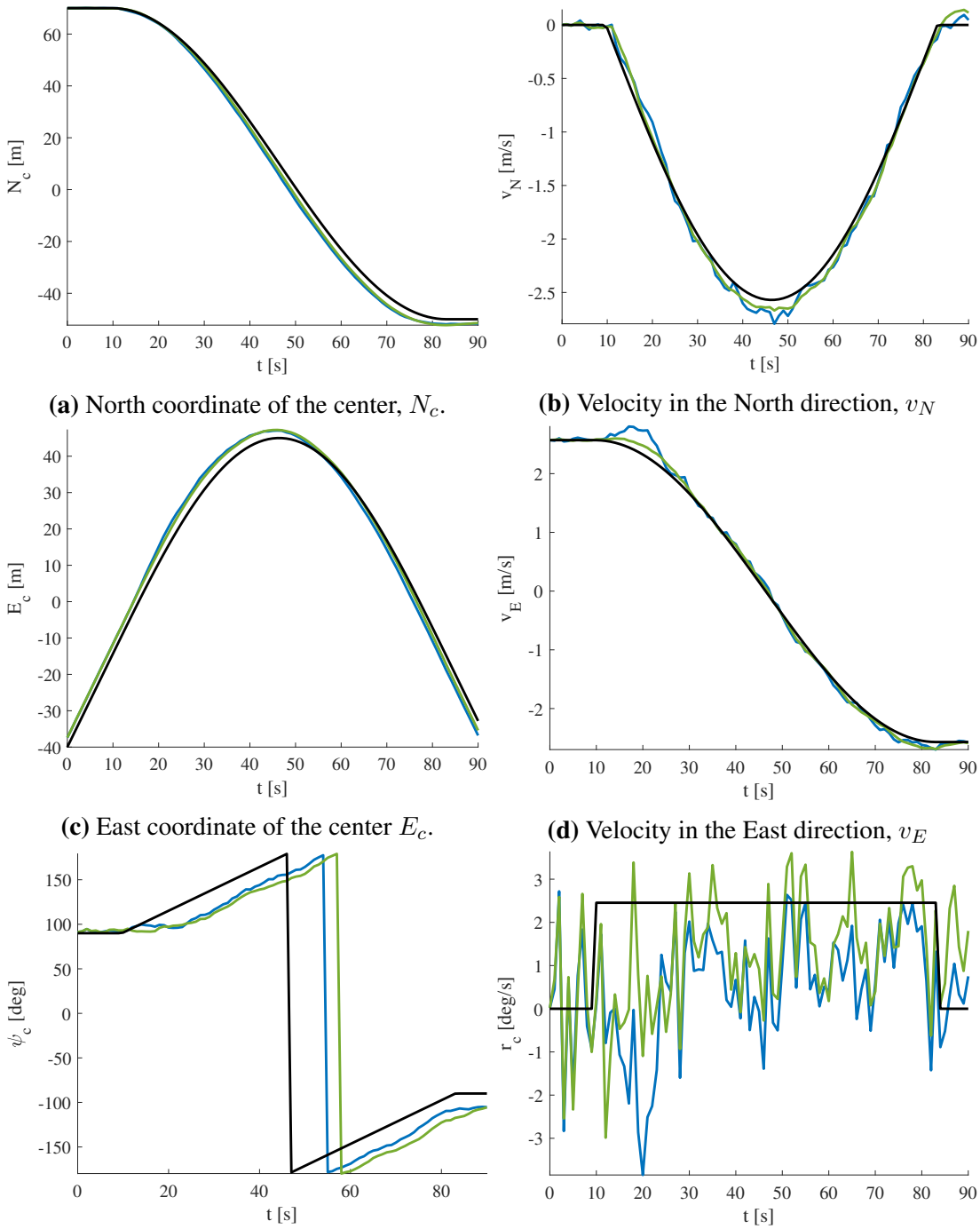


(a) $t_k = 1\,\text{s}$. **(b)** $t_k = 12\,\text{s}$. **(c)** $t_k = 23\,\text{s}$.

(d) $t_k = 34\,\text{s}$. **(e)** $t_k = 46\,\text{s}$. **(f)** $t_k = 57\,\text{s}$.

(g) $t_k = 68\,\text{s}$. **(h)** $t_k = 79\,\text{s}$. **(i)** $t_k = 90\,\text{s}$.

Figure 8.15: Snapshots for different time points of the lidar measurements (red), the true hull position (black) and the estimated hull position using the measurement model with explicit differentiation (blue) and with implicit differentiation (green). The solid blue and green lines correspond to the current estimate, while the dashed lines correspond to the estimate for the previous time point. The estimated positions of the center are shown as isolated blue and green circles, while the blue and green circles correspond to predict measurements.



(a) North coordinate of the center, N_c.

(b) Velocity in the North direction, v_N

(c) East coordinate of the center E_c.

(d) Velocity in the East direction, v_E

(e) Heading angle, ψ_c.

(f) Angular rate, r_c.

Figure 8.16: True state values (black) and estimated values using the measurement model with explicit differentiation (blue) and with implicit differentiation (green).