

EXTENDED TARGET TRACKING USING REDUCED RANK GAUSSIAN  
PROCESSES

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MUSTAFA BUĞRA ÖZCAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2021



Approval of the thesis:

**EXTENDED TARGET TRACKING USING REDUCED RANK GAUSSIAN PROCESSES**

submitted by **MUSTAFA BUĞRA ÖZCAN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar  
Dean, Graduate School of **Natural and Applied Sciences** \_\_\_\_\_

Prof. Dr. İlkey Ulusoy  
Head of Department, **Electrical and Electronics Engineering** \_\_\_\_\_

Assoc. Prof. Dr. Emre Özkan  
Supervisor, **Electrical and Electronics Eng. Dept., METU** \_\_\_\_\_

**Examining Committee Members:**

Prof. Dr. Umut Orguner  
Electrical and Electronics Engineering Department, METU \_\_\_\_\_

Assoc. Prof. Dr. Emre Özkan  
Electrical and Electronics Engineering Department, METU \_\_\_\_\_

Prof. Dr. Çağatay Candan  
Electrical and Electronics Engineering Department, METU \_\_\_\_\_

Assist. Prof. Dr. Mustafa Mert Ankaralı  
Electrical and Electronics Engineering Department, METU \_\_\_\_\_

Assist. Prof. Dr. Osman Serdar Gedik  
Computer Engineering Dept., Ankara Yıldırım Beyazıt Uni. \_\_\_\_\_

**Date:** 12.02.2021

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: Mustafa Buğra Özcan

Signature :

## **ABSTRACT**

### **EXTENDED TARGET TRACKING USING REDUCED RANK GAUSSIAN PROCESSES**

Özcan, Mustafa Buğra

M.S., Department of Electrical and Electronics Engineering

Supervisor : Assoc. Prof. Dr. Emre Özkan

February 2021, 88 pages

Conventional tracking algorithms are predominantly based on point target assumption; however, this assumption is challenged as a result of the advents in sensor resolutions. Improvements on processors and rapid advances in sensor capabilities has enabled to the perception of target characteristics beyond the kinematics. Extended target tracking is the ability to learn target shapes that occupy multiple resolution cells and to track the motion of the target in a recursive framework. Gaussian process, a non-parametric method to settle the bridge between inputs and outputs of a system with tractable math, is opted for this thesis to establish the extended target tracking structure. On the other hand, computational complexity can be considered as a cost of the flexible nature of Gaussian processes. We, therefore, investigate the periodic kernel spectral approximation that relies on restating the kernel matrix with a lower rank. Following this, we perform in-depth mathematical derivations, conduct simulations in comparison with other approaches, and discuss several aspects of our

extended target tracking method.

Keywords: Extended Target Tracking, Object Tracking, Gaussian Process, Kalman Filter

## ÖZ

### **DERECESİ AZALTILMIŞ GAUSS SÜREÇLERİ KULLANILARAK GENİŞLETİLMİŞ HEDEF TAKİBİ**

Özcan, Mustafa Buğra

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Doç. Dr. Emre Özkan

Şubat 2021, 88 sayfa

Geleneksel takip algoritmaları büyük ölçüde nokta hedef varsayımına bağlı olmasına rağmen, bu varsayımın yeterliliği sensor çözünürlüklerindeki gelişmeler neticesinde sorgulanmaya başlanmıştır. İşlemcilerde gerçekleştirilen ilerlemeler ve sensör kapasitelerindeki hızlı artışlar, hedeflerin kinematik özelliklerinin ötesinde şekilsel karakteristiklerini de algılayabilmeye imkan sağlamıştır. Genişletilmiş hedef takibi birden fazla çözünürlük hücrelerine yayılan cisimlerin şeklini yinelemeli olarak öğrenme ve hareketini takip etme yeteneğidir. Bir sistemin girdi ve çıktıları arasındaki köprüyü parametrelere bağlı kalmadan, kavraması zor olmayan bir matematikle oluşturabilen Gauss süreçleri, bu tezdeki genişletilmiş hedef takibi yapısını kurmak için tercih edilmiştir. Diğer bir taraftan Gauss süreçlerinin doğasındaki bu esnekliğin bedeli olarak işlem karmaşıklığı göz önünde bulundurulmalıdır. Bu problemin çözümüne periyodik kernel matrislerinin daha düşük dereceli olarak ifade edilebilen spektral yakınsaması ile zemin hazırlanmıştır. Bununla birlikte detaylı matematiksel çıkarımlar yapılmış,

diđer metodlar ile karşılaştırmalı benzetimler gerçekleştirilmiş ve metodun incelikleri tartışılmıştır.

Anahtar Kelimeler: Genişletilmiş Hedef Takibi, Obje Takibi, Gauss Süreci, Kalman Filtresi

*Without the wind, waves can not go beyond the rocks...*

*To my family and my w(l)ife...*

## ACKNOWLEDGMENTS

It is an immense pleasure to convey my profound sense of joy and gratitude to the people who, in one way or another, contributed to the completion of this prosperous thesis.

In the first place, I would like to express my humble regards and esteemed thanks to my revered supervisor Assoc. Prof. Dr. Emre Özkan for his privileged comments, insightful suggestions, and for giving me the freedom to develop my own research path.

I am candidly thankful for the METU Sensor Fusion Group, especially for Murat. I have always remembered our fruitful discussions on the Gaussian processes, target tracking, and sensor fusion.

I am also earnestly grateful to ASELSAN Inc. for supporting me in my graduate education. I also wish to thank for the strong sense of camaraderie within my colleagues and friends.

I would like to extend my heartfelt thanks to both my family and my in-laws as they have always supported me by all means in my whole life. I bestow my unfathomable reverence and dutiful obeisance to my mother for her enthusiastic inspiration. The abundant attention and care of my brother Hakan towards me have helped me through bumpy rides on numerous occasions.

Last but foremost, I feel substantial thanks to my best and most precious discovery of life, my wife Kübra, for her efforts in supporting all my endeavors, care to cherish within our moods, and her monumental color to life.

## TABLE OF CONTENTS

ABSTRACT . . . . .	v
ÖZ . . . . .	vii
ACKNOWLEDGMENTS . . . . .	x
TABLE OF CONTENTS . . . . .	xi
LIST OF TABLES . . . . .	xiv
LIST OF FIGURES . . . . .	xv
LIST OF ALGORITHMS . . . . .	xviii
LIST OF ABBREVIATIONS . . . . .	xix
CHAPTERS	
1 INTRODUCTION . . . . .	1
2 BAYESIAN FILTERING . . . . .	7
2.1 Bayesian Filtering . . . . .	7
2.1.1 Prediction Update . . . . .	8

2.1.2	Measurement Update . . . . .	9
2.2	Kalman Filter . . . . .	10
2.3	Extended Kalman Filter . . . . .	12
3	GAUSSIAN PROCESSES . . . . .	15
3.1	Kernel Functions . . . . .	16
3.1.1	Exponential Kernel . . . . .	16
3.1.2	Squared Exponential Kernel . . . . .	17
3.1.3	Periodic Kernel . . . . .	19
3.2	Hyperparameters . . . . .	19
3.3	Gaussian Process Regression . . . . .	20
3.4	Approximations of GP Regression . . . . .	25
4	REDUCED RANK GAUSSIAN PROCESS REGRESSION . . . . .	27
4.1	Reduced Rank Approximation . . . . .	27
4.1.1	Basis Functions with Dirichlet Boundary Conditions	28
4.1.2	Basis Functions with Periodic Boundary Conditions	32
4.1.3	Prediction with Reduced Rank Kernel Function . .	36
4.2	Recursive Form . . . . .	39

4.3	Learning Target Extent Using Reduced Rank Gaussian Process Regression . . . . .	44
5	AUGMENTED STATE SPACE MODEL FOR EXTENDED TARGET TRACKING . . . . .	51
5.1	Proof of the Concept . . . . .	57
5.2	Computational Aspects . . . . .	62
5.3	Performance Evaluations . . . . .	67
6	CONCLUDING REMARKS AND FUTURE WORK . . . . .	77
	REFERENCES . . . . .	81

## LIST OF TABLES

### TABLES

Table 4.1 Equations for radius with respect to the angle which starts from the center of the shape [15] . . . . .	45
Table 5.1 Comparison table for measurement update steps in ETT-GP and ETT-RRGP . . . . .	62
Table 5.2 Significant time complexities in ETT-GP for different variables compared to ETT-RRGP. $\mathcal{O}(\cdot)^*$ shows the complexity due to matrix inverse operations, which can be dismissed by precalculations and storing in memory. The number of bases is denoted as $m$ , and $n$ refers to the number of measurements. . . . .	64
Table 5.3 Number of operations concerning significant terms for typical 2-D and 3-D scenarios in comparison with ETT-GP and ETT-RRGP . . . . .	65
Table 5.4 Time complexities in EKF measurement update equations introduced in equations 2.12a to 2.12d. $m$ stands for the state dimension, and $n$ is the number of measurements. . . . .	65
Table 5.5 RMSE and IOU to compare ETT-GP and ETT-RRGP algorithms for a square extent along a diagonal path . . . . .	71
Table 5.6 RMSE and IOU to compare ETT-GP and ETT-RRGP algorithms for a triangle extent along a U-turn path . . . . .	71

## LIST OF FIGURES

### FIGURES

Figure 1.1 Different modeling approaches for the same target . . . . .	4
Figure 1.2 (a) is a video experiment data [40] and (b) shows 2D laser data of a vehicle together with the extent estimates [66]. . . . .	5
Figure 1.3 A 3D laser scanner data in [42]. (a) and (b) are camera views and (c) shows the laser sensor data together with the extent estimates. . . . .	5
Figure 3.1 Exponential kernel with $\sigma^2 = 1$ and $l = 2$ . . . . .	17
Figure 3.2 Squared exponential kernel with $\sigma^2 = 1$ and $l = 2$ . . . . .	18
Figure 3.3 Periodic kernel with $\sigma^2 = 1, l = 2$ and $p = 2\pi$ . . . . .	20
Figure 3.4 $K(x, x')$ using a periodic kernel with $\sigma^2 = 1, l = 2$ and $p = 2\pi$ . . .	21
Figure 3.5 SE kernel with different hyperparameters . . . . .	22
Figure 3.6 Illustration of a GP regression example. The dashed line is the unknown function, plus signs are the measurements, and the continuous line shows the mean prediction with 95% confidence interval area. . . . .	24
Figure 3.7 A 3D visualization of the GP regression example . . . . .	25
Figure 4.1 A visualization of a few bases in the interval $[0, 2\pi]$ with Dirichlet boundary conditions . . . . .	32

Figure 4.2	A 2D illustration of the product $\tilde{\Phi} = \Phi\Phi^T$ of a few bases in the interval $[0, 2\pi]$ with Dirichlet boundary conditions . . . . .	33
Figure 4.3	A visualization of a few bases in the interval $[0, 2\pi]$ . . . . .	36
Figure 4.4	Prediction with reduced rank GP with Dirichlet boundary conditions and full GP SE kernel solution . . . . .	40
Figure 4.5	Prediction with reduced rank GP with periodic boundary conditions and full GP periodic kernel solution . . . . .	41
Figure 4.6	An arbitrary star domain with an example line segment from $\zeta_0$ to $\zeta$ . . . . .	44
Figure 4.7	An illustration of target extent representations for a peanut-like shape in different domains . . . . .	46
Figure 4.8	An illustration of target extent representations for a square shape in different domains . . . . .	46
Figure 4.9	Recursive reduced rank GP prediction of a peanut-like shape with the squared exponential kernel in different time instances . . . . .	47
Figure 4.10	Recursive reduced rank GP prediction of a peanut-like shape with the squared exponential kernel . . . . .	48
Figure 4.11	Recursive reduced rank GP prediction of a peanut-like shape with the periodic kernel . . . . .	49
Figure 5.1	Coordinate systems representations of an extended target . . . . .	52
Figure 5.2	Different coordinate systems of the same extended target at two consecutive time instances as $k_0$ and $k_1$ . Red cross signs represent the measurements as $l_0$ and $l_1$ belonging to each time . . . . .	53
Figure 5.3	An example ETT result of a peanut-like shape by using Algorithm 5.1 (a) General view with a few sampled time instances, (b) Enlarged result of near initial time, (c) Magnified final time instance . . . . .	60

Figure 5.4 Estimation and ground truth graphics for positions and orientation for visualization in Figure 5.3 . . . . .	61
Figure 5.5 Average computation times per update in simulations. . . . .	66
Figure 5.6 An example case with RMSE control points for the target extent . . . . .	68
Figure 5.7 Intersection over union illustration for extended targets . . . . .	69
Figure 5.8 Example illustrations for different IOU levels . . . . .	69
Figure 5.9 An example ETT illustration for a diagonal path with a square shape using ETT-GP and ETT-RRGP with sampled time instances . . . . .	70
Figure 5.10 An example ETT illustration for a U-turn path with a triangle shape using ETT-GP and ETT-RRGP with sampled time instances . . . . .	72
Figure 5.11 A general panorama of IOU difference between ETT-RRGP and ETT-GP with respect to the number of measurements and bases . . . . .	73
Figure 5.12 A randomly sampled extent for periodic kernel . . . . .	73
Figure 5.13 Average IOU results for model matched targets using ETT-RRGP, ETT-GP, and full GP methods with indicators for 15 and 80 bases on each comparison. Indicators emphasize that the results for the same computa- tion time per update with different algorithms. . . . .	74

## LIST OF ALGORITHMS

### ALGORITHMS

Algorithm 4.1	Batch regression with reduced rank GP . . . . .	38
Algorithm 4.2	Recursive regression with reduced rank GP . . . . .	43
Algorithm 5.1	Extended Target Tracking with Reduced Rank Gaussian Pro- cesses . . . . .	58

## LIST OF ABBREVIATIONS

### ABBREVIATIONS

GP	Gaussian Process
KF	Kalman Filter
EKF	Extended Kalman Filter
PDF	Probability Density Function
SE	Squared Exponential
RMSE	Root Mean Square Error
IOU	Intersection over Union
RRGP	Reduced Rank Gaussian Process
ETT	Extended Target Tracking
MC	Monte Carlo
SSM	State-Space Model
SLAM	Simultaneous Localization and Mapping
$\mathcal{N}$	Normal Distribution
MLE	Maximum Likelihood Estimation



## CHAPTER 1

### INTRODUCTION

As Albert Einstein summed up, "We cannot solve our problems with the same thinking we used when we created them." This can inspire us to think that mathematical models have been enhanced by recent growth in technology. From the Wiener filter to the Kalman filter and beyond, researchers have been improving existing techniques for arising problems. Estimation theory, an important keystone of them, deals with the unknown in problems based on observational data. Numerous dynamical systems consist of these unknown quantities as they have a random component in the form of time series. A mathematical framework for this implicit relation between the unknown, observations, and time can be described with state-space models (SSMs). They are considered as tailored models to reveal statistically coherent estimations. Object tracking involves estimation of unknown kinematic state variables of moving objects, which are modeled as dynamical systems. In typical target tracking algorithms, states are selected to represent the target's kinematic features such as position, velocity, or acceleration.

Bayesian filtering is a framework to obtain posterior probability density function (PDF) of the state given the measurements. Kalman filter (KF) instantiates Bayesian filtering for linear systems corrupted by Gaussian noise with a closed-form solution. KFs can be used in tracking applications when the underlying dynamics and the measurement equation is assumed to be linear. Kalman filter is beneficial when PDFs are unimodal with linear interactions. For nonlinear systems, it is possible to use Kalman filter variants such as extended Kalman filter (EKF), which involves linear approximation [29], and unscented Kalman filter (UKF), which relies on sigma points [31].

When the degree of nonlinearity is severe, or the posterior PDF is multimodal, more sophisticated algorithms can be used, such as Gaussian sum filtering [3] or particle filters [20, 35, 28].

In the history of target tracking, the objects of interest have been modeled as point sources [13, 4, 71]. This assumption is valid for most sensors such as radar, sonar, or lidar, which generate at most one measurement per scan. Over the last two decades, rapid advances in sensor capability and improvements on processors have opened the door for extended target tracking (ETT) algorithms [22]. These dedicated tracking algorithms are capable of processing multiple measurements per object and estimate the object's extent, which can be defined as the region that generates measurements.

There are several challenges, such as computational burden, changes in object orientation, unknown target shape to overcome in the ETT problem. The target model is an important factor related to the challenges [22, 45]. Various attempts have been carried out on modeling the target extent. The first investigations into ETT consider target shape as restricted constant models. Studies are conducted to model the extent as a stick in [19, 14] and more recently in [6, 23]. The main pitfall of that approach is the dimension of the extent. It fails to explain the cross extent of the target, which is perpendicular to the line. The extent can never be more than one dimension with this approach.

The close resemblance between a vehicle and a rectangle leads to ETT studies to restrict the shape with four right angles [24, 36]. On the other hand, target extent is represented as a circle in [11]. Moreover, modeling the shape as an ellipsoid gives more flexible restrictions and increases the variety of applications [37, 18]. An inference method and a comparison between ellipsoid or rectangle modeling are investigated in [24]. Ellipsoid modeling, which is carried out by a symmetric positive semi-definite matrix, is also a common solution [48, 43]. Several studies suggested an extension of ellipsoid modeling to multi-ellipsoids for representing more complex shapes [33, 27, 25]. These models divide targets into several parts and represent each of them as an ellipse to capture more details about the target. However, in all of these suggestions initializing with a restricted boundary for the shape can be considered

as a drawback of these methods. Another fold of studies, star convex shapes, are suggested in [8] and can be a solution to this restriction. An initial version of this method introduced in [37] is also compared by the study in [7] and further discussed in [62, 9]. A related suggestion is proposed in [34] by utilizing B-splines to express vehicle contour. These instruments adopted in the field of ETT are comprehensively surveyed in [22]. The aforementioned methods are illustrated comparatively in Figure 1.1. In the figure, the same target is represented using different approaches.

A method well-suited to the star convex shapes using Gaussian processes (GPs) is proposed in [66]. GPs occupy a significant place in machine learning applications and are extensively covered in [54]. They provide tractable posterior computations and have useful analytical properties. The model in [66] presents systematic expressions for tracking the target extent and has been adopted by several research papers [1, 41, 65]. A 3D implementation is suggested in [41], and classification works are carried out in [47, 65].

GPs are also utilized for learning ambient magnetic fields [58], and later for simultaneous localization and mapping (SLAM) algorithms [39]. Although GP can fulfill decent solutions to the regression problem, its characteristics should be taken into account in terms of computational complexity, recursive structures, and closed-form solutions. The spectral domain solutions, particularly reduced rank forms of GP regression, have received much attention over the last years [58, 39]. This form is opted for our work due to its outstanding results in finite space solutions and the ability to be formulated in a recursive form. Related works consider other applications and only examine squared exponential kernel with Dirichlet boundary conditions [30, 58, 39]. We refined the method for periodic functions and proposed a systematic structure for ETT applications.

Autonomous driving, mobile robotics, or computer vision can be possible application areas for the method provided in this thesis. Figure 1.2 shows ETT examples with real data for a video experiment of a dinghy [40] and a 2D laser experiment of a vehicle [66]. Another example ETT application is depicted in Figure 1.3 for a 3D traffic scenario studied in [42]. Our method can also be used for similar applications.

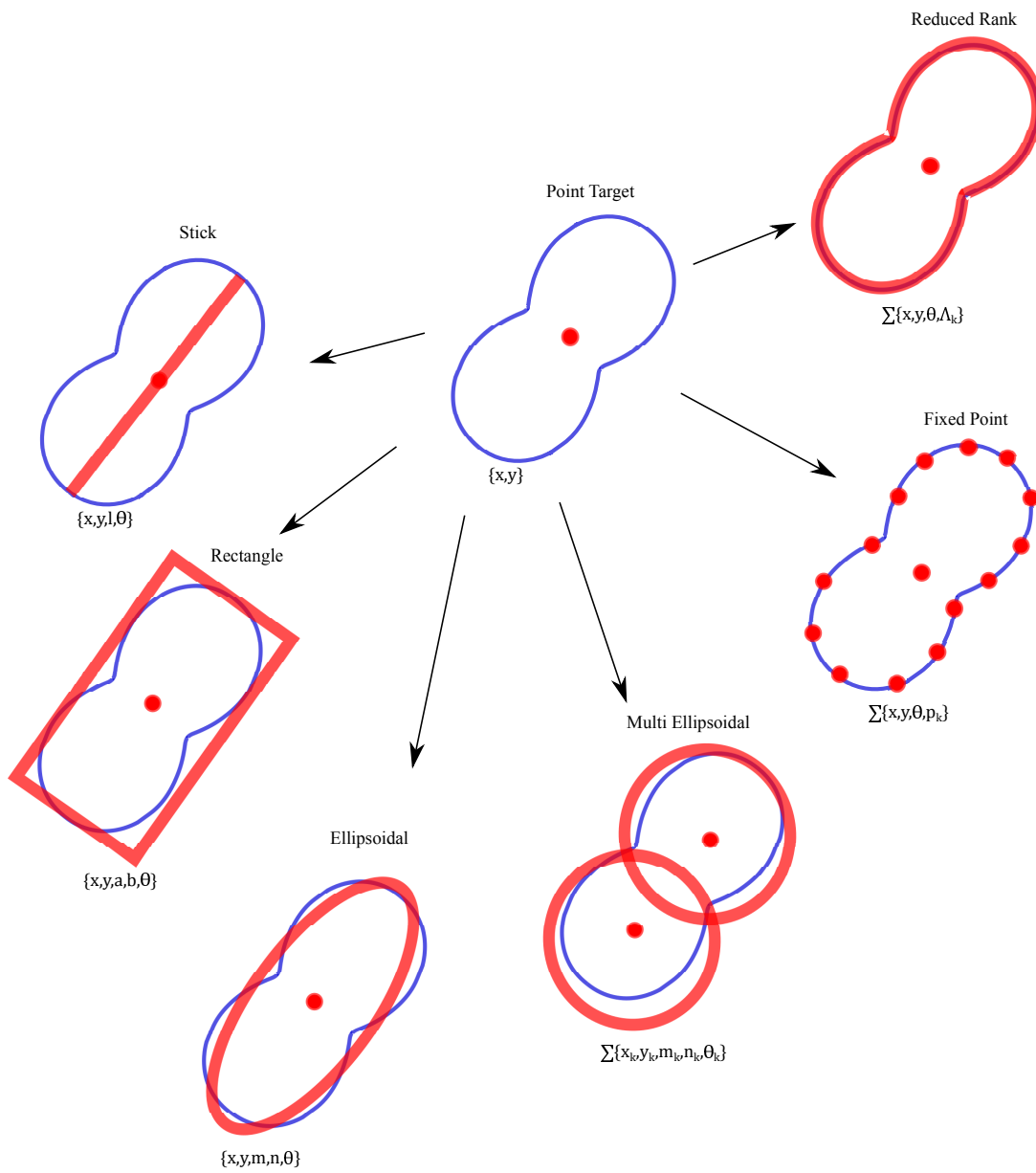


Figure 1.1: Different modeling approaches for the same target

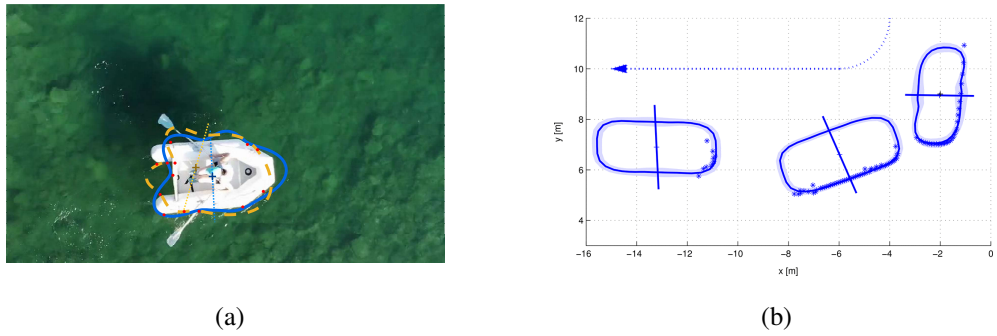


Figure 1.2: (a) is a video experiment data [40] and (b) shows 2D laser data of a vehicle together with the extent estimates [66].

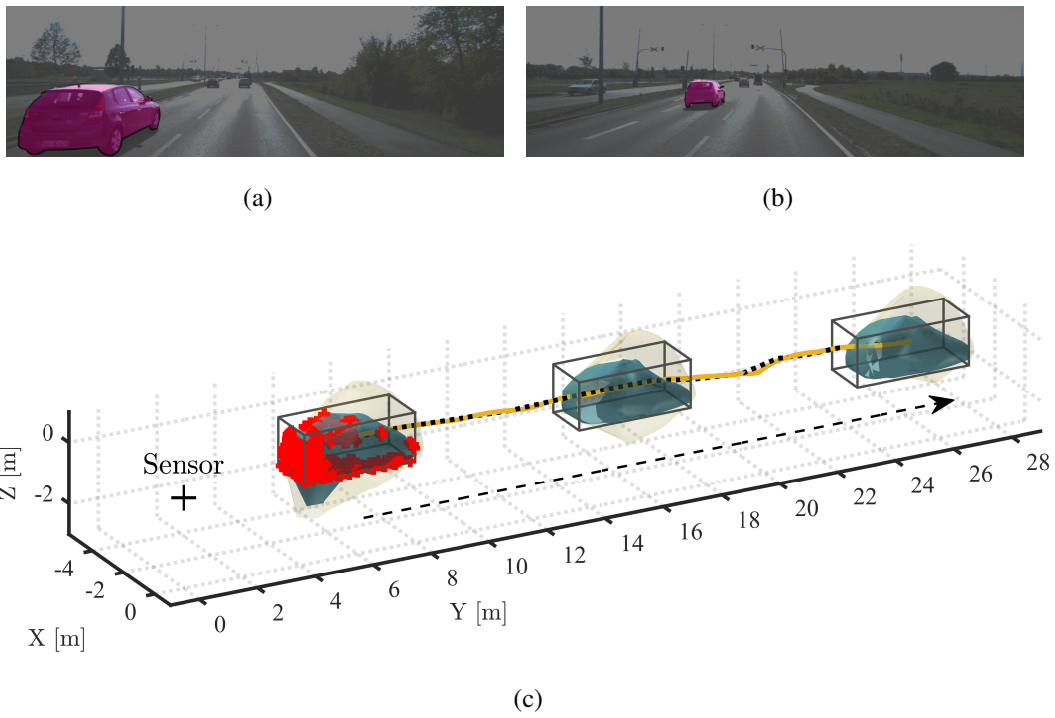


Figure 1.3: A 3D laser scanner data in [42]. (a) and (b) are camera views and (c) shows the laser sensor data together with the extent estimates.

The thesis is composed of six chapters. Chapter 2 presents the theoretical principles of Bayesian filtering, its closed-form solution under certain conditions, and reviews Kalman filter and its variants for nonlinear system dynamics. Random processes are also renowned for the myriad of applications, and they are introduced in Chapter 3. The reduced rank approximation is given in Chapter 4 with explicit derivations, and a learning method of a target shape is presented. In Chapter 5, augmented SSM is asserted with proposed algorithms for ETT applications. Finally, a brief conclusion is outlined in Chapter 6.

## CHAPTER 2

### BAYESIAN FILTERING

This chapter aims to address the fundamental principles of target tracking. Although it has concise information about Bayesian filtering, Kalman filter, and extended Kalman filter, derivations are also included to have a good grasp.

Establishing a mathematical framework is the primary component of a target tracking algorithm. State-space models (SSMs) constitute the backbone of the tracking algorithms as they construct the algebraic structure that defines target dynamics and measurement equations. The core problem is to obtain the posterior distribution of the state after compounding measurements. Different methods for computing posterior distributions are addressed in the following sections.

#### 2.1 Bayesian Filtering

Bayesian filtering can be defined as a Bayesian method of optimum filtering [64]. It formulates marginal posterior distribution of the state given the measurements in terms of products of probability density functions. State can be very briefly explained as a summary of the past. In target tracking applications, it may consist of information about target position, velocity, orientation, or any other variable to be estimated. A general state-space model can be described with two equations

$$x_k = f(x_{k-1}, w_{k-1}), \quad (2.1a)$$

$$y_k = g(x_k, v_k), \quad (2.1b)$$

for  $k = 1, 2, \dots$ , where

- $x_k \in \mathbb{R}^m$  is the state vector at time  $k$ ,
- $y_k \in \mathbb{R}^n$  is the measurement available at time  $k$ ,
- $f(\cdot)$  is the known system equation generally describing the model dynamics,
- $g(\cdot)$  is the known measurement equation which defines the relation between the state and measurements,
- $w_{k-1}$  is the process noise,
- $v_k$  is the measurement noise.

When measurements are acquired up to the  $k^{\text{th}}$  time instance, Bayesian filtering aims to obtain the posterior probability density function of the state  $p(x_k | y_{1:k})$ . This is mostly performed in recursive updates in two steps, which are called the prediction and measurement updates.

### 2.1.1 Prediction Update

Prediction update is the process of finding the state's probability density function for a future time step given the available measurements. The aim of the prediction update is to obtain  $p(x_k | y_{1:k-1})$  by calculating the integral

$$p(x_k | y_{1:k-1}) = \int p(x_k, x_{k-1} | y_{1:k-1}) d_{x_{k-1}}, \quad (2.2a)$$

$$= \int p(x_k | x_{k-1}, y_{1:k-1}) p(x_{k-1} | y_{1:k-1}) d_{x_{k-1}}. \quad (2.2b)$$

This equation is known as the Chapman-Kolmogorov equation in the literature. It relates joint PDFs by employing the law of total probability [51]. Furthermore, with the assumption that the current state only depends on the previous state, which is known as Markov property, it is possible to express the prediction density as

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1}) p(x_{k-1}|y_{1:k-1}) d_{x_{k-1}}. \quad (2.3)$$

Attempts to solve this integral equation might be difficult, depending on the relevant distributions. Under linear relations with Gaussian assumptions, more convenient forms can be achieved.

### 2.1.2 Measurement Update

When a new measurement is acquired, new posterior distribution can be calculated by applying the Bayes rule using the result obtained in the prediction update. This step aims to include the information that comes with the new measurement to the prediction density. Given all measurements up to time  $k$ , the posterior of the state variable can be obtained as follows

$$\begin{aligned} p(x_k|y_{1:k}) &= p(x_k|y_k, y_{1:k-1}), \\ &= \frac{p(x_k, y_k, y_{1:k-1})}{p(y_k, y_{1:k-1})}, \\ &= \frac{p(y_k|x_k, y_{1:k-1}) p(x_k, y_{1:k-1})}{p(y_k, y_{1:k-1})}, \\ &= \frac{p(y_k|x_k, y_{1:k-1}) p(x_k|y_{1:k-1}) p(y_{1:k-1})}{p(y_k|y_{1:k-1}) p(y_{1:k-1})}. \end{aligned} \quad (2.4)$$

This equation can be further simplified by assuming that measurement  $y_k$  is conditionally independent of  $y_l$  for  $l < k$  given  $x_k$  such that

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k) p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}. \quad (2.5)$$

Many researchers have attempted to assert different methods for the calculation of these products. Not only do they try to find an approximate equation depending on the distribution properties, but they also derive compact analytical formulas under certain conditions.

## 2.2 Kalman Filter

With R.E. Kalman's seminal paper in 1960, "A New Approach to Linear Filtering and Prediction Problems" [32], Kalman filter (KF) is a frequently used technique today. It provides the closed-form optimal solution of the filtering problem in equation 2.1 under linear Gaussian assumptions, which indicate that  $f(\cdot)$  and  $g(\cdot)$  are linear functions and the noise terms are Gaussian. A linear Gaussian SSM can be expressed as

$$x_k = Ax_{k-1} + w_{k-1}, \quad (2.6a)$$

$$y_k = Hx_k + v_k, \quad (2.6b)$$

for  $k = 1, 2, \dots$ , where

- $x_k \in \mathbb{R}^m$  is the state vector at time  $k$ ,
- $y_k \in \mathbb{R}^n$  is the measurement available at time  $k$ ,
- $A$  is a  $m \times m$  matrix representing linear relationship between states for successive time instances,
- $H$  is a  $n \times m$  matrix to relate the state with the measurement,
- $w_{k-1} \sim \mathcal{N}(0, Q)$ ,  $Q$  is the process noise covariance matrix,
- $v_k \sim \mathcal{N}(0, R)$ ,  $R$  is the measurement noise covariance matrix.

In this characteristic KF SSM representation, prior state distribution is also assumed as Gaussian. Hence, the posterior and predictive distributions can be expressed as follows

$$p(x_k|y_{1:k}) = \mathcal{N}(x_k; m_{k|k}, P_{k|k}), \quad (2.7a)$$

$$p(x_k|y_{1:k-1}) = \mathcal{N}(x_k; m_{k|k-1}, P_{k|k-1}), \quad (2.7b)$$

where the mean of the resulting PDF is defined as  $m$ , and covariance of it is referred as  $P$ . Kalman filter solutions can be obtained in the same manner as the Bayesian filtering equations introduced in Section 2.1.

In the prediction update, the posterior density obtained from the previous time step is propagated in time according to system dynamics to find the PDF of the state in the next time instance. Moreover, the aim of the measurement update is to utilize new measurement information for the best possible posterior PDF. Hence, well-known KF update equations can be expressed as follows:

- Prediction Update:

$$m_{k|k-1} = Am_{k-1|k-1}, \quad (2.8a)$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q. \quad (2.8b)$$

- Measurement Update:

$$S_k = HP_{k|k-1}H^T + R, \quad (2.9a)$$

$$K_k = P_{k|k-1}H^T S_k^{-1}, \quad (2.9b)$$

$$m_{k|k} = m_{k|k-1} + K_k (y_k - Hm_{k|k-1}), \quad (2.9c)$$

$$P_{k|k} = P_{k|k-1} - K_k H P_{k|k-1}. \quad (2.9d)$$

Where  $K_k$  and  $S_k$  are referred to as Kalman gain and innovation matrix, respectively.

### 2.3 Extended Kalman Filter

Linear system dynamics or linear measurement state relations often may not occur in practical applications. In such situations, classical KF equations cannot be used. An approximation method should be employed to overcome this problem. Extended Kalman filter is capable of handle this nonlinearity. It utilizes Taylor series expansion to obtain the posterior distribution. Consider the following nonlinear SSM with additive Gaussian noise

$$x_k = f(x_{k-1}) + w_{k-1}, \quad (2.10a)$$

$$y_k = h(x_k) + v_k. \quad (2.10b)$$

for  $k = 1, 2, \dots$ , where

- $x_k \in \mathbb{R}^m$  is the state at time  $k$ ,
- $y_k \in \mathbb{R}^n$  is a measurement obtained at time  $k$ ,
- $f(\cdot)$  is a nonlinear function representing the relationship between states for successive time instances, and it might also depend on  $k$ ,
- $h(\cdot)$  is a nonlinear function to relate the state with the measurement, and it might also depend on  $k$ ,
- $w_{k-1} \sim \mathcal{N}(0, Q)$ ,  $Q$  is the process noise covariance matrix,
- $v_k \sim \mathcal{N}(0, R)$ ,  $R$  is the measurement noise covariance matrix.

EKF linearizes dynamic and/or the measurement equations to obtain posterior PDF of the state recursively. It uses first-order terms in Taylor series expansion and neglects the higher-order terms. So, EKF relies on the Jacobian matrix with the help of the predicted state because the exact state is not available. Although EKF has some drawbacks, such as dependency on the Jacobian matrix, it is one of the commonly used

tracking algorithms in terms of computational efficiency, ease of use, and theoretical stability. EKF equations can be written as follows:

- Prediction Update:

$$m_{k|k-1} = f(m_{k-1|k-1}), \quad (2.11a)$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q. \quad (2.11b)$$

- Measurement Update:

$$S_k = H_k P_{k|k-1} H_k^T + R, \quad (2.12a)$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1}, \quad (2.12b)$$

$$m_{k|k} = m_{k|k-1} + K_k (y_k - h(m_{k|k-1})), \quad (2.12c)$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1}, \quad (2.12d)$$

where Jacobians can be written as:

$$F_k = \left. \frac{\partial f}{\partial x} \right|_{x_{k-1|k-1}}, \quad (2.13a)$$

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{x_{k|k-1}}. \quad (2.13b)$$

Throughout this chapter Bayesian filtering equations are introduced, and KF, a closed-form solution under certain conditions, is presented. Unlike the KF that is suited for linear models, there are other filtering variants for nonlinear systems. EKF, which is considered as one of them, is adopted in the works for this thesis. Since they are fundamental inference methods that are related to the subject of our work, a brief background is outlined. Another principal subject of our research, which is Gaussian processes, is examined in the next chapter.



## CHAPTER 3

### GAUSSIAN PROCESSES

Gaussian Processes can be considered as a powerful tool to learn a function by computing posterior distribution with observed measurements, especially for machine learning applications. Apart from being solely a stochastic process, it is a non-parametric method to establish a bridge between inputs and outputs with tractable math. Since the prior distribution of functions can be interpreted as GP, the posterior distribution can be predicted by defining correlation structures in a probabilistic approach. The closeness of function values with respect to inputs is induced by a kernel. It is a key component for GP and will be explained in Section 3.1.

A latent function with a GP prior can be denoted as

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')), \quad (3.1)$$

where  $m(x)$  is the mean function, and  $k(x, x')$  is the kernel function. Since they are sufficient statistics for the Gaussian distribution, multiple input evaluation can be written for points  $\mathbf{x} = [x_1, x_2, \dots, x_L]$  as follows

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_L) \end{bmatrix} \sim \mathcal{N}(\mathbf{m}(\mathbf{x}), K(\mathbf{x}, \mathbf{x})), \quad (3.2)$$

where

$$\mathbf{m}(\mathbf{x}) = \begin{bmatrix} m(x_1) \\ \vdots \\ m(x_L) \end{bmatrix}, \quad (3.3a)$$

$$K(\mathbf{x}, \mathbf{x}) = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_L) \\ \vdots & \ddots & \vdots \\ k(x_L, x_1) & \dots & k(x_L, x_L) \end{bmatrix}. \quad (3.3b)$$

### 3.1 Kernel Functions

A kernel function is an essential factor in constructing a GP prior. It is a way of expressing our belief about the unknown function. While we do not know the latent function, we might have an intuition about some characteristics, such as smoothness, periodicity, or any knowledge about the correlation between either inputs or function values. A straightforward assumption might be that closer input points tend to have similar function values so that neighboring cells give more information about the underlying function value at the desired point. This assumption mathematically requires the kernel to be a function of  $x - x'$ . The term stationary kernel has been used to refer to this specification. Moreover, if it is a function of  $\|x - x'\|$ , which is the norm distance between input points, the kernel is termed isotropic. There exists a variety of kernels that manage different properties; hence, it should be chosen by "kernel engineering". The choice of a kernel depends on the target function behavior. Some prominent examples are given in the following subsections.

#### 3.1.1 Exponential Kernel

The exponential kernel defines the correlation between two points by considering an exponential function. Since it is a function of  $|x - x'|$ , it is both stationary and isotropic. The exponential kernel is denoted as

$$k(x, x') = \sigma^2 \exp\left(-\frac{|x - x'|}{l}\right). \quad (3.4)$$

For this representation,  $l$  is called the length scale, which shows characteristics of correlation over distance, and  $\sigma^2$  is the scale parameter that can be thought as correlation amplitude. The distance can be tuned with the length scale parameter, and correlation can be adjusted with the scale parameter. An example of the exponential kernel with  $\{\sigma^2 = 1, l = 2\}$  can be seen in Figure 3.1 as a function of distance.

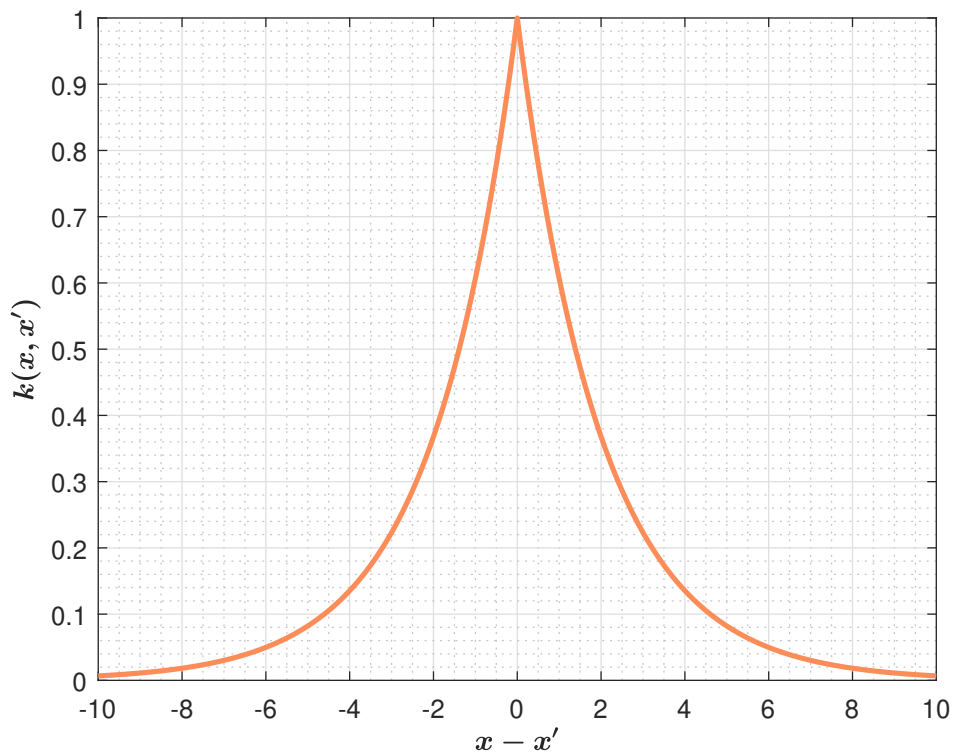


Figure 3.1: Exponential kernel with  $\sigma^2 = 1$  and  $l = 2$

### 3.1.2 Squared Exponential Kernel

Squared exponential (SE) kernel, exponentiated quadratic kernel, is one of the favored kernel functions. Not only does it ensure a smoother correlation line, but it also has a closed-form spectral density equation. Although the level of smoothness may be

undesirable in certain applications in which this high smoothness is not suitable, there are also many successful works using SE kernel. It is considered as a default kernel for a variety of applications [54], [68]. It can be constructed as follows

$$k(x, x') = \sigma^2 \exp\left(-\frac{|x - x'|^2}{2l^2}\right). \quad (3.5)$$

The scale parameter denoted as  $\sigma^2$ , and the length scale parameter, denoted as  $l$ , are used for the same purpose in all kernels. The former is used to tune correlation amplitude, and the latter is used to control the correlation length. An example is given in Figure 3.2 in comparison with the exponential kernel.

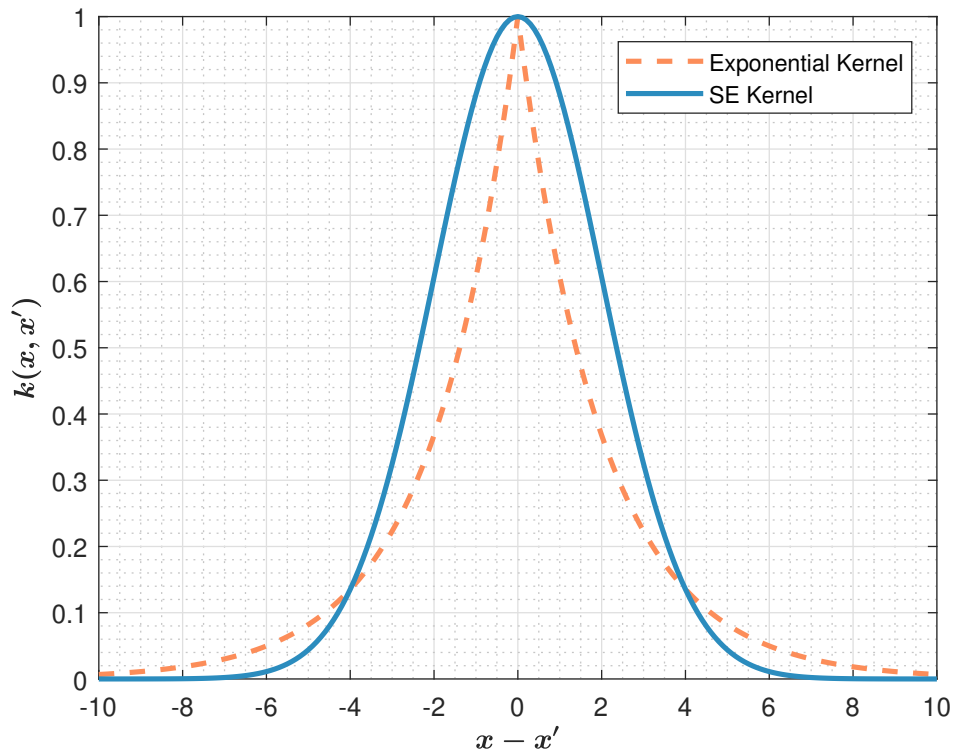


Figure 3.2: Squared exponential kernel with  $\sigma^2 = 1$  and  $l = 2$

### 3.1.3 Periodic Kernel

Periodicity can be encountered in a lot of real-world observations as well as in mathematical or physical models. Therefore, some latent functions may repeat a pattern between subsequent identical intervals as a mirror of this phenomenon. In order to adopt this behavior to the GP framework, periodic kernels are used. An example periodic kernel can be expressed as

$$k(x, x') = \sigma^2 \exp \left( -\frac{2}{l^2} \sin^2 \left( \frac{\pi |x - x'|}{p} \right) \right), \quad (3.6)$$

where  $p$  is a parameter that expresses assumed periodicity,  $\sigma^2$  and  $l$  are other hyperparameters to adjust amplitude and length scale. A periodic kernel is shown in Figure 3.3. Furthermore, a visualization of its matrix form, which is introduced in equation 3.3a can be seen in Figure 3.4 with a period of  $2\pi$ .

## 3.2 Hyperparameters

There might be a question that although GP is called a non-parametric model, how some parameters are frequently mentioned up to this section. A model can be described as a parametric method if the number of parameters representing the function can be compressed into a finite-dimensional parameter vector [54]. Unlike this definition, GP only parameterizes the distribution over functions, and the term hyperparameters are used to emphasize that they are indeed tuning parameters of the kernel function [46].

In order to thoroughly learn the unknown function, hyperparameters should be tuned for addressing the function behavior. Hyperparameters can be determined in a brute force method, or some particular estimation method, e.g., maximum likelihood estimation (MLE) can be used. More details about the MLE technique are stated in [54] with problems similar to multiple local optimums and some solutions. Figure 3.5 demonstrates the effect of different hyperparameters for SE kernel concerning the

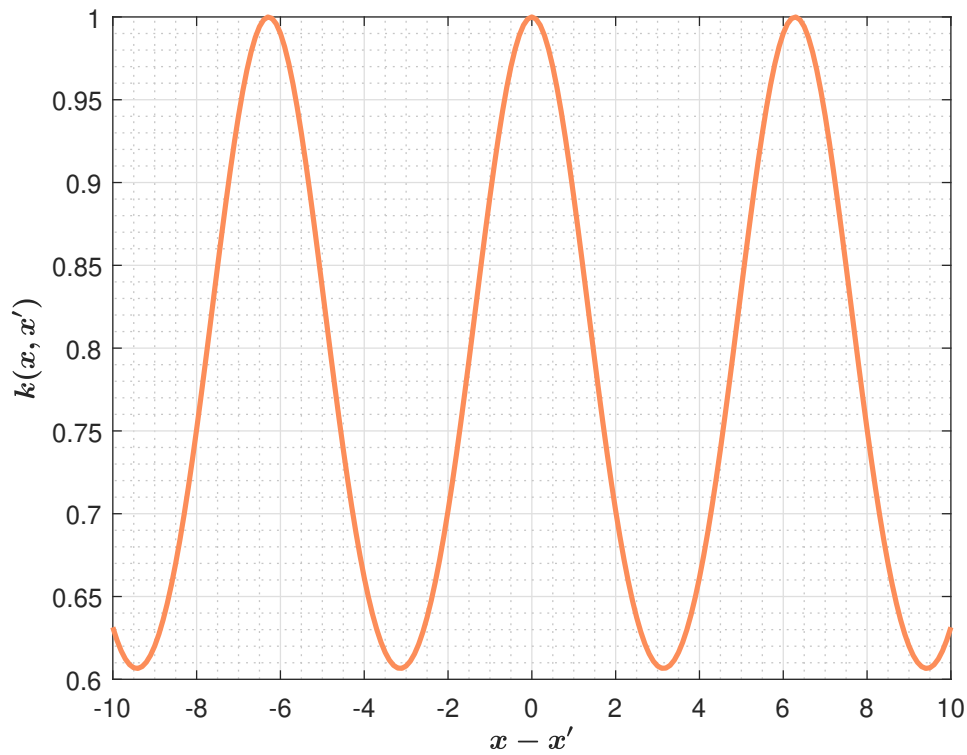


Figure 3.3: Periodic kernel with  $\sigma^2 = 1$ ,  $l = 2$  and  $p = 2\pi$

distance between inputs.

### 3.3 Gaussian Process Regression

Regression is the problem of predicting a continuous quantity of a certain outcome. The regression problem can be solved by GPs. One of the advantages of the method is that GP regression knows how much it knows [54]. It can predict a point estimate with its uncertainty metric. More specifically, since GP constructs a model where the function is jointly Gaussian, the marginal distribution of a new point, which is also a Gaussian, has a covariance value. If the standard regression problem can be formulated by assuming a prior for the function, it is formulated as follows

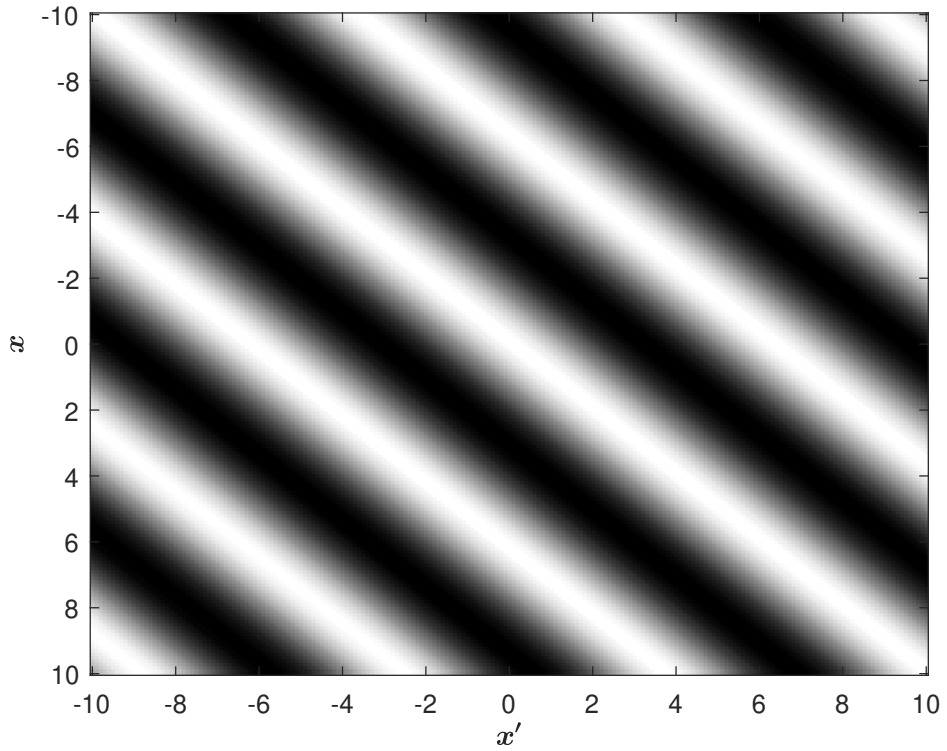


Figure 3.4:  $K(x, x')$  using a periodic kernel with  $\sigma^2 = 1, l = 2$  and  $p = 2\pi$

$$z = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_r^2), \quad (3.7)$$

where

- $z$  is the observed measurement corrupted with noise,
- $f(\cdot)$  is the unknown function modeled as GP as in equation 3.1,
- $\epsilon$  is the Gaussian measurement noise with zero mean and variance  $\sigma_r^2$ .

This problem can be expanded for multiple observation case with multivariate Gaussian distribution, as introduced in equation 3.2. Multi-dimensional notation can be denoted as

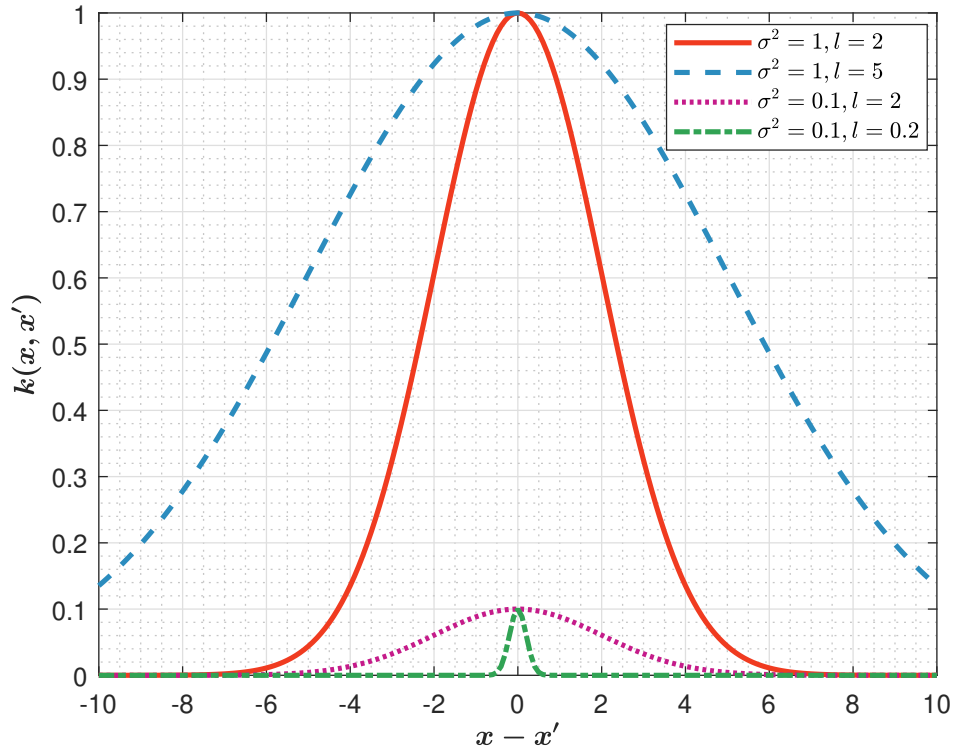


Figure 3.5: SE kernel with different hyperparameters

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{f}(\mathbf{x}^*) \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \mathbf{m}(\mathbf{x}) \\ \mathbf{m}(\mathbf{x}^*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{x}, \mathbf{x}) + I_N \otimes \sigma_r^2 & K(\mathbf{x}, \mathbf{x}^*) \\ K(\mathbf{x}^*, \mathbf{x}) & K(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix} \right), \quad (3.8)$$

where

- $\mathbf{x}$  is the vector of  $L$  points where measurements are collected and represented as  $\mathbf{x} \triangleq [x_1 \dots x_L]^\top$ ,
- $\mathbf{z}$  is the set of measurements  $\mathbf{z} \triangleq [z_1, \dots, z_L]^\top$ ,
- $\mathbf{x}^*$  consist of the location of test inputs such that  $\mathbf{x}^* \triangleq [x_1^*, \dots, x_L^*]^\top$ ,
- $\mathbf{f}(\mathbf{x}^*)$  is the desired function values at the test inputs  $\mathbf{x}^*$ ,
- $\sigma_r^2$  is the covariance for measurements,

- $\mathbf{m}(\mathbf{x})$  and  $\mathbf{m}(\mathbf{x}^*)$  are the mean vectors of marginal distributions for corresponding inputs such that  $\mathbf{m}(\mathbf{x}) \triangleq [m(x_1), \dots, m(x_L)]^\top$  and  $\mathbf{m}(\mathbf{x}^*) \triangleq [m(x_1^*), \dots, m(x_L^*)]^\top$ ,
- $K(\cdot)$  is the kernel function introduced in equation 3.3a.

Consequently, the marginal distribution of the desired function values  $p(\mathbf{f}(\mathbf{x}^*)|\mathbf{z})$  can be calculated as another Gaussian,  $p(\mathbf{f}(\mathbf{x}^*)|\mathbf{z}) \sim \mathcal{N}(\mathbf{m}^*, \Sigma^*)$ . The posterior mean  $\mathbf{m}^*$  and posterior covariance  $\Sigma^*$  values can be written with the two equations below

$$\mathbf{m}^* = \mathbf{m}(\mathbf{x}^*) + K(\mathbf{x}^*, \mathbf{x}) [K(\mathbf{x}, \mathbf{x}) + I_N \otimes \sigma_r^2]^{-1} (\mathbf{z} - \mathbf{m}(\mathbf{x})), \quad (3.9a)$$

$$\Sigma^* = K(\mathbf{x}^*, \mathbf{x}^*) - K(\mathbf{x}^*, \mathbf{x}) [K(\mathbf{x}, \mathbf{x}) + I_N \otimes \sigma_r^2]^{-1} K(\mathbf{x}, \mathbf{x}^*). \quad (3.9b)$$

Usually, prior of the mean function,  $\mathbf{m}(\mathbf{x})$  is chosen as 0 for most of the applications. However, this choice is not such a restrictive assumption so that the posterior mean could be anything depending on the underlying function. Moreover, for notational brevity, some intermediate variables can be defined such that

$$S = K(\mathbf{x}^*, \mathbf{x}) K_{\mathbf{z}}^{-1}, \quad (3.10a)$$

$$P = K(\mathbf{x}^*, \mathbf{x}^*) - SK(\mathbf{x}, \mathbf{x}^*), \quad (3.10b)$$

$$K_{\mathbf{z}} = K(\mathbf{x}, \mathbf{x}) + I_N \otimes \sigma_r^2. \quad (3.10c)$$

Hence, mean and covariance functions of the test points can be restated as  $\mathbf{m}^* = S\mathbf{z}$  and  $\Sigma^* = P$ .

An illustration of the GP regression is visualized in Figure 3.6 and Figure 3.7 with the following properties,

- A quintic function, which is described as a fifth-degree polynomial, is chosen as the unknown function,

- $L = 5$  measurements are observed at locations  $\mathbf{x} = [-0.5 \ 0.2 \ 1.5 \ 2 \ 2.5]^\top$  with measurement noise variance  $\sigma_r^2 = 0.05$ ,
- Hyperparameters are set to  $\sigma^2 = 5$  and  $l = 0.5$ ,
- Test inputs are defined very dense; whereas, two specific yet arbitrary locations are determined to draw the posterior PDFs in Figure 3.7.

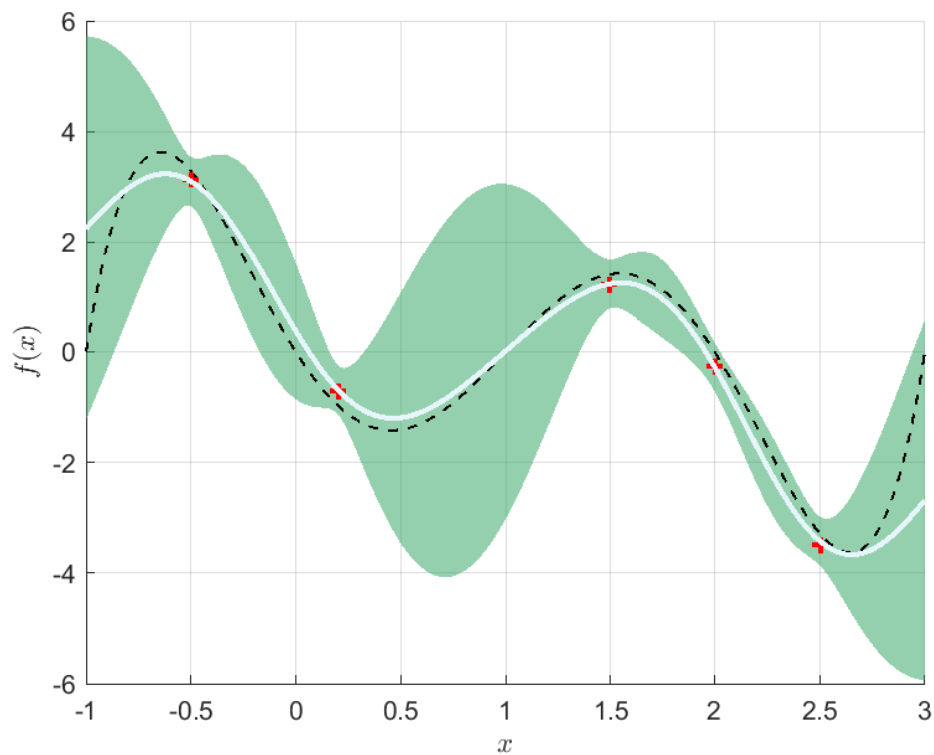


Figure 3.6: Illustration of a GP regression example. The dashed line is the unknown function, plus signs are the measurements, and the continuous line shows the mean prediction with 95% confidence interval area.

One of the significant advantages of it is knowing the confidence interval about prediction, which can be proved by investigating intervals with no measurements. It is higher at less observed locations and lower at neighboring measurement locations.

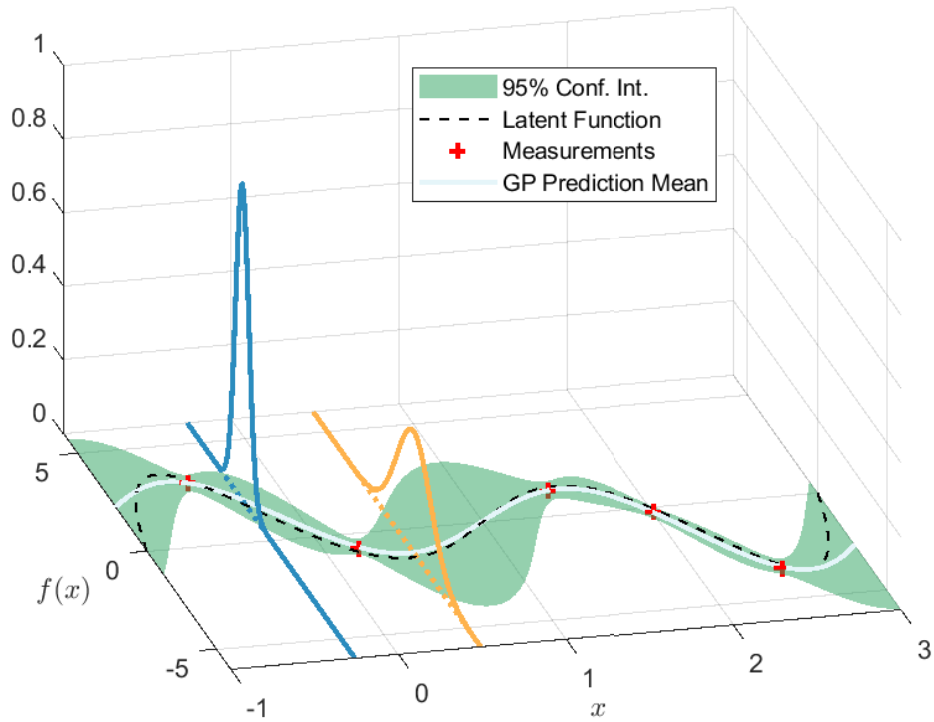


Figure 3.7: A 3D visualization of the GP regression example

### 3.4 Approximations of GP Regression

Despite the fact that GP regression can be utilized for Bayesian regression, it may suffer from computational complexity. This drawback may be interpreted as a cost of the non-parametric nature of GP regression. This complexity is particularly important when investigating big data sets. It can be seen in equation 3.9 that Kernel functions account for this burden while computing matrix inverse. This problem can be outlined as requiring  $\mathcal{O}(n^3)$  operations for inversion, where  $n$  is the number of measurements.

While exploiting the GP regression, this shortcoming is seen to be a possible dilemma. Therefore, a growing body of literature has been proposed to overcome this issue [16], [55], [56], [26]. These methods can be interpreted as two approaches regarding the solution domain in which approximation is performed. The first approach can be stated as utilizing pseudo inputs to construct kernel function and relating them with

real inputs. Despite the fact that the way of the regulating locations of pseudo inputs or their usage in equations changes, they assume pseudo variables and represent the solution in the time domain where GP literally lies. On the other hand, recently, there has been considerable interest in spectral representation of the kernel function [59]. When this approximation is considered, real input values are exploited instead of pseudo ones, and one of them is entirely pointed out in Section 4. Besides, many studies are published on comparing sparse approximations [5], [53].

## CHAPTER 4

### REDUCED RANK GAUSSIAN PROCESS REGRESSION

When considering GP as a flexible tool that provides solutions to the regression problem, real-time applications have to be taken into account in terms of computational complexity, recursive structures, and closed-form solutions. Over the last decade, many researchers have studied these issues by developing sparse solutions aforementioned in Section 3.4. There are also other approaches that aim to find a solution in the spectral domain. Particularly, the reduced rank form of GP regression is addressed in [58], [30], and [39]. This form is opted for this thesis due to its convenience for finite space problems and its ability to be formulated in a recursive structure. Even though comprehensive studies are carried out, to the best of our knowledge, related applications only examine squared exponential kernel and Dirichlet boundary conditions. We refined the method for periodic functions, and this aspect is presented throughout the chapter.

#### 4.1 Reduced Rank Approximation

This method relies on restating the kernel function with lower rank but enabling equations to be written algebraically more tractable. This optimum approximation is mentioned in [54]; however, the use of that form is very inconvenient due to computing eigendecomposition operation. It offers no remedy with regard to the exact GP solution. Fortunately, for restricted domains, eigenfunctions can be calculated in a more efficient way. It is thoroughly studied in [59] with extensive details and comparisons. The approximation is based on constructing the kernel function in the following way

$$k(x, x') \approx \sum_j S(\sqrt{\lambda_j}) \phi_j(x) \phi_j(x'), \quad (4.1)$$

where  $S(\cdot)$  is the spectral density of the kernel, and it can be calculated either numerically or in closed-form.  $\phi_n(\cdot)$  is  $n^{\text{th}}$  basis in a set that spans the desired space associated with  $\lambda_n$  and may have a closed-form expression for the input argument.

Some other works do not include closed-form expressions due to the complexity of the considered space, such as [39] for a hexagonal domain or [57] for more sophisticated domains that might be non-convex or disconnected. While they cannot state an explicit function for bases, the numerical calculation of bases is also possible by storing them in memory. In our study, closed-form expressions are given in Section 4.1.2 for the domain in which we solved the ETT problem.

It is presented that bases can be found by solving an eigenvalue problem for the Laplacian operator for a finite domain [59]. In contrast to standard cases, our problem involves a periodic function. In order to satisfy periodicity, the eigenvalue problem for the Laplacian operator should be solved by imposing different boundary conditions.

Solutions of this approach can be obtained by solving a second-order differential equation, which is explicitly derived in the next sections with different boundary conditions.

#### 4.1.1 Basis Functions with Dirichlet Boundary Conditions

Dirichlet Boundary Condition stands for specifying a constant value at the edges of the space. It is predominantly used in many areas, from fluid mechanics [12] to nuclear physics [21]. In statistics, several studies, such as [59], [58], or [39], have been carried out with solutions of Dirichlet boundary conditions through a finite domain. In this section, explicit solutions of the Laplace equation with respect to Dirichlet boundary conditions are given. The problem can be restated with the following

piecewise-defined function with 0 at the boundary

$$\begin{cases} -\nabla^2 \phi_n(x) = \lambda_n \phi_n(x), & x \in \Omega, \\ \phi_n(x) = 0, & x \in \partial\Omega. \end{cases} \quad (4.2)$$

It can also be written as a second-order differential equation such that

$$\frac{d^2 \phi_n(x)}{dx^2} + \lambda_n \phi_n(x) = 0, \quad 0 < x < L, \quad (4.3)$$

with following limitations at the boundary of the domain

$$\phi_n(0) = \phi_n(L) = 0. \quad (4.4)$$

The methodology which is performed in the nonlinear physical science book [67] as separations of variables is broadened for non-trivial solutions of this type of second-order differential equations into 3 cases, which are  $\lambda_n < 0$ ,  $\lambda_n = 0$ , and  $\lambda_n > 0$ .

- For  $\lambda_n < 0$  :

If it is denoted as  $\lambda_n = -\omega^2$ , the differential equation can be written in the following form

$$\phi_n''(x) - \omega^2 \phi_n(x) = 0. \quad (4.5)$$

General solutions of this kind of equations are also examined in [60] and defined as

$$\phi_n(x) = c_1 e^{\omega x} + c_2 e^{-\omega x}. \quad (4.6)$$

Applying the boundary constraint  $x(0) = 0$  into the general solution results in

$$c_2 = -c_1. \quad (4.7)$$

Substituting this coefficient gives the form

$$\phi_n(x) = 2c_1 \sinh(wx). \quad (4.8)$$

Moreover, the other condition  $x(L) = 0$  requires

$$2c_1 \sinh(wL) = 0. \quad (4.9)$$

This equation forces the  $c_1 = 0$ ; therefore, it does not have a non-trivial solution for  $\lambda_n < 0$ .

- For  $\lambda_n = 0$  :

General solution becomes

$$\phi_n(x) = c_1 + c_2x, \quad (4.10)$$

as also stated in [60]. Applying  $x(0) = 0$  gives

$$c_1 = 0, \quad (4.11)$$

and the other condition  $x(L) = 0$  allows writing the coefficient

$$c_2 = 0. \quad (4.12)$$

Hence, the equation becomes  $\phi_n(x) = 0$ ; therefore, there is not a non-trivial solution for this case.

- For last case which is  $\lambda_n > 0$  :

If we again write  $\lambda_n = \omega^2$ , the differential equation can be written as

$$\phi_n''(x) + \omega^2 \phi_n(x) = 0. \quad (4.13)$$

General solutions of this kind of equations are also given in [60] as

$$\phi_n(x) = c_1 \cos(\omega x) + c_2 \sin(-\omega x). \quad (4.14)$$

The boundary condition at 0 makes  $c_1 = 0$ . By applying the other boundary condition to the resulting solution, it can be written as

$$c_2 \sin(wL) = 0. \quad (4.15)$$

Besides, apart from the trivial solution that  $c_2 = 0$ , non-trivial solutions can be the roots of

$$\sin(wL) = 0. \quad (4.16)$$

Hence, a solution can be written for  $\omega$  as  $\frac{n\pi}{L}$  for  $n = 1, 2, \dots$ . Therefore

$$\lambda_n = \left(\frac{n\pi}{L}\right)^2 \quad \text{where } n = 1, 2, \dots, \quad (4.17)$$

and corresponding eigenfunctions are

$$\sin\left(\frac{2n\pi x}{L}\right) \quad \text{where } n = 1, 2, \dots \quad (4.18)$$

These forms coincide with previous works [59], [58]. It should also be noted that forcing bases to a strict value at boundaries might have some deficiency. Function values may deviate from true values due to reaching these boundary conditions near the end of the domain. In order to overcome this issue, basis functions are evaluated with a safety margin than desired such that a larger domain is calculated. After this evaluation, a subset of it that stands for the actual domain can be used. Figure 4.1 demonstrates a few bases obtained in the above equations in interval  $L = 2\pi$ . It also pinpoints the boundary values apparently at the edges of the graph.

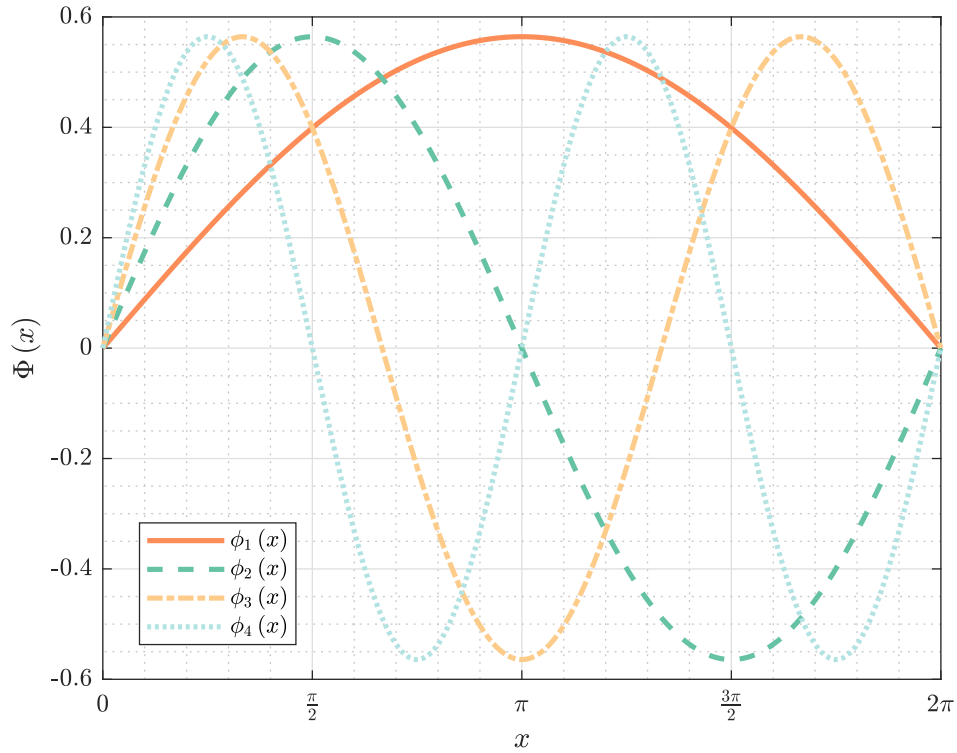


Figure 4.1: A visualization of a few bases in the interval  $[0, 2\pi]$  with Dirichlet boundary conditions

### 4.1.2 Basis Functions with Periodic Boundary Conditions

Although the ETT problem actually needs a periodic function, a periodicity assumption is not presented yet. Problem statements in other works, as for instances [59, 58, 30, 39] can be restated for periodic boundary conditions such that

$$\left\{ \begin{array}{l} -\nabla^2 \phi_n(x) = \lambda_n \phi_n(x), x \in \Omega \\ \phi_n(x), x \in \partial\Omega \\ 0, x \notin \Omega \end{array} \right. \quad (4.19)$$

In this section, the explicit solution of equation 4.19 is given by writing it as a second-order differential equation similar to two dimensional heat equation in [70] as follows:

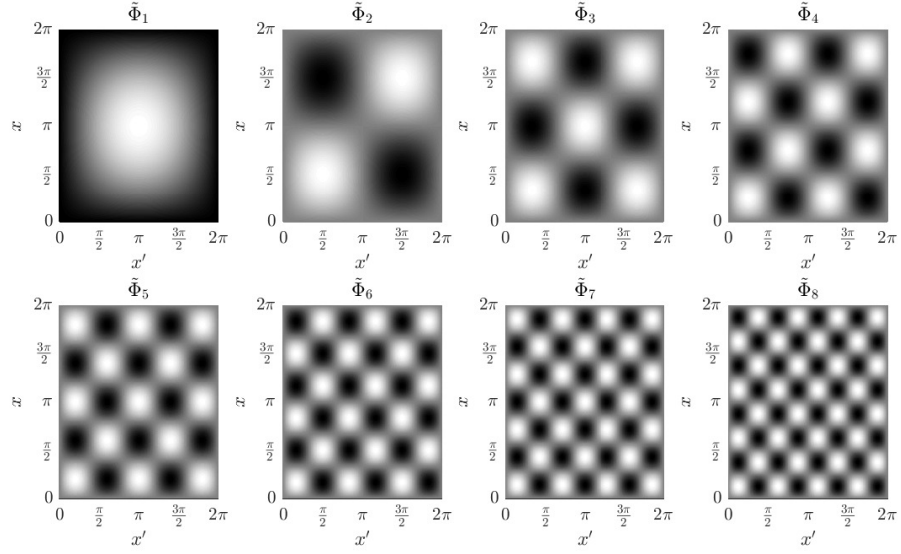


Figure 4.2: A 2D illustration of the product  $\tilde{\Phi} = \Phi\Phi^T$  of a few bases in the interval  $[0, 2\pi]$  with Dirichlet boundary conditions

$$\frac{d^2\phi_n(x)}{dx^2} + \lambda_n\phi_n(x) = 0, \quad 0 < x < L \quad (4.20)$$

In order to achieve a periodic pattern, boundary conditions are designated with the following two identities

$$\phi_n(0) = \phi_n(L), \quad (4.21a)$$

$$\phi'_n(0) = \phi'_n(L). \quad (4.21b)$$

A complementary methodology is used to obtain closed-form final solutions analogous to [67]. Second-order differential equations are broadly investigated in [60]. Non-trivial solutions of this second-order differential equation are examined in 3 cases, which are  $\lambda_n < 0$ ,  $\lambda_n = 0$ , and  $\lambda_n > 0$ .

- For  $\lambda_n < 0$  :

If it is denoted that  $\lambda_n = -\omega^2$ , the differential equation can be written as

$$\phi_n''(x) - \omega^2 \phi_n(x) = 0. \quad (4.22)$$

General solutions of this kind of equations are given in [60]

$$\phi_n(x) = c_1 e^{\omega x} + c_2 e^{-\omega x}. \quad (4.23)$$

Using boundary constraints, equations are obtained such that

$$\begin{aligned} c_1 + c_2 &= c_1 e^{\omega L} + c_2 e^{-\omega L}, \\ c_1 - c_2 &= c_1 e^{\omega L} - c_2 e^{-\omega L}. \end{aligned} \quad (4.24)$$

After algebraic manipulations, the resulting equation can be written as

$$e^{\omega L} = e^{-\omega L}. \quad (4.25)$$

Since this equation is only satisfied for  $\omega = 0$ , there is no solution for  $\lambda_n < 0$ .

- For  $\lambda_n = 0$  :

The general solution becomes [60],

$$\phi_n(x) = c_1 + c_2 x. \quad (4.26)$$

Afterward, using boundary constraints,  $c_2$  can be found as 0, and  $c_1$  can be found as  $\phi_n(x)$ . So for  $\lambda_n = 0$ , one eigenfunction can be found as 1

- For the last case, which is  $\lambda_n > 0$  :

If we again write  $\lambda_n = \omega^2$ , the differential equation can be written as

$$\phi_n''(x) + \omega^2 \phi_n(x) = 0. \quad (4.27)$$

General solutions of this kind of equations are [60],

$$\phi_n(x) = c_1 \cos(\omega x) + c_2 \sin(-\omega x). \quad (4.28)$$

Using boundary constraints, we can obtain equations such that

$$\begin{aligned} c_1 &= c_1 \cos(\omega L) + c_2 \sin(-\omega L), \\ c_2 &= c_2 \cos(\omega L) - c_1 \sin(-\omega L). \end{aligned} \quad (4.29)$$

After algebraic calculations, It can be written that,

$$\sin(\omega L) = 0 \quad \text{and} \quad \cos(\omega L) = 1. \quad (4.30)$$

Hence,  $\omega$  can be found as  $\frac{2n\pi}{L}$  for  $n = \pm 1, \pm 2, \dots$ . Therefore

$$\lambda_n = \left(\frac{2n\pi}{L}\right)^2 \quad \text{where} \quad n = \pm 1, \pm 2, \dots, \quad (4.31)$$

and corresponding eigenfunctions:

$$\cos\left(\frac{2n\pi x}{L}\right) \quad \text{and} \quad \sin\left(\frac{2n\pi x}{L}\right) \quad \text{where} \quad n = \pm 1, \pm 2, \dots \quad (4.32)$$

To conclude, the solution to the equation 4.19 can be written for  $m$  basis function as

$$\begin{cases} \lambda_0 = 0 \quad \text{and} \quad \phi_0(x) = 1 \quad \text{for} \quad n = 0 \\ \lambda_n = \left(\frac{2n\pi}{L}\right)^2 \quad \text{and} \quad \phi_n(x) = \cos\left(\frac{2n\pi x}{L}\right), \quad \sin\left(\frac{2n\pi x}{L}\right) \quad \text{for} \quad n = \pm 1, \pm 2, \dots \end{cases} \quad (4.33)$$

These forms correlate favorably with previous findings for similar structures in the literature [70]. A visualization of a few basis functions is given in Figure 4.3. Furthermore, it should be pointed out that basis functions are independent of hyperparameters of the kernel function. Hyperparameters can be adjusted with the spectral representation of the kernel. Besides, it should be taken into account that when the number of basis points defined as  $m$  and domain boundary  $L$  goes to infinity, approximation reaches to exact kernel function as mentioned in [30] and proved in [59].

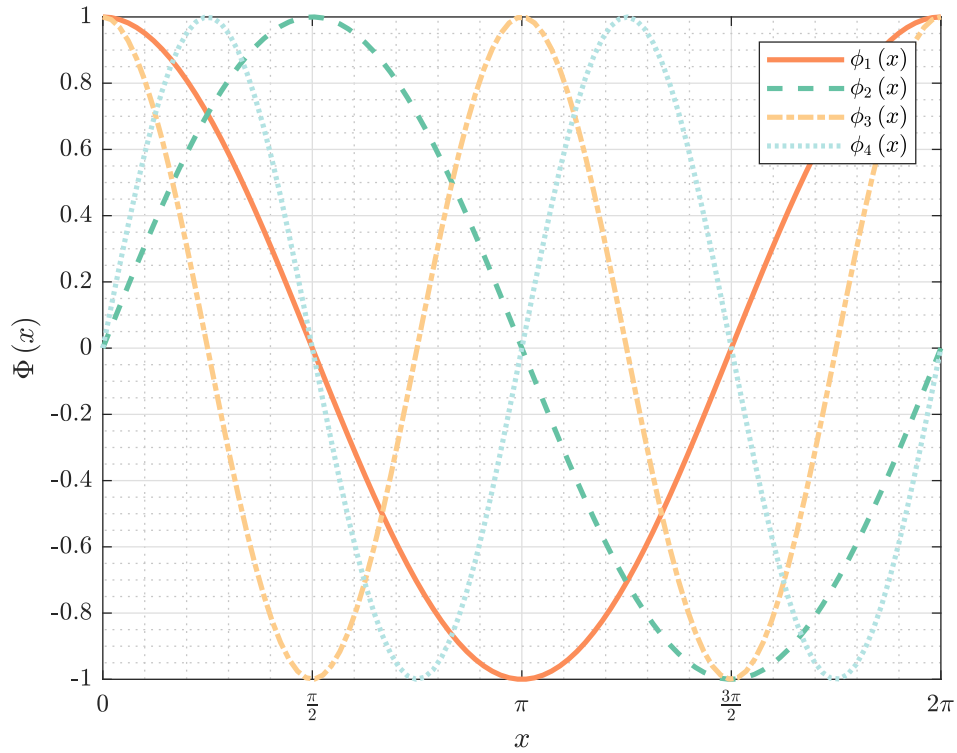


Figure 4.3: A visualization of a few bases in the interval  $[0, 2\pi]$

### 4.1.3 Prediction with Reduced Rank Kernel Function

In this section, the advantages of using the reduced rank kernel function are shown with final form equations. The matrix structure of the low rank kernel function introduced in equation 4.1 can be denoted as

$$K(\mathbf{x}, \mathbf{x}) \approx \Phi \Lambda \Phi^\top, \quad (4.34)$$

where

- $K(\mathbf{x}, \mathbf{x}) \in R^{n \times n}$  is the kernel matrix for the input vector, introduced in equation 3.2,
- $\Phi \in R^{n \times m}$  matrix with the rows  $\Phi_i$ ,

- $\Phi_i$  is a row vector corresponding for the inputs  $x_i$  where  $i = 1, 2, \dots, n$  such that  $\Phi_i = [\phi_1(x_i), \phi_2(x_i), \dots, \phi_m(x_i)]$ ,
- $\Lambda$  is a diagonal  $m \times m$  matrix whose terms are presented in equation 4.1.

Analogously,  $\Phi_*$  can be described in the same way for the test input  $x^*$ , and kernel matrix forms are given as

$$K(\mathbf{x}, \mathbf{x}^*) \approx \Phi \Lambda \Phi_*^\top, \quad (4.35a)$$

$$K(\mathbf{x}^*, \mathbf{x}) \approx \Phi_* \Lambda \Phi^\top, \quad (4.35b)$$

$$K(\mathbf{x}^*, \mathbf{x}^*) \approx \Phi_* \Lambda \Phi_*^\top. \quad (4.35c)$$

Substituting these kernel matrix forms into the exact expression, which is given in equation 3.9, one can write

$$\mathbf{m}^* = \mathbf{m}(\mathbf{x}^*) + \Phi_* \Lambda \Phi^\top [\Phi \Lambda \Phi^\top + I_N \otimes \sigma_r^2]^{-1} (\mathbf{z} - \mathbf{m}(\mathbf{x})), \quad (4.36a)$$

$$\Sigma^* = \Phi_* \Lambda \Phi_*^\top - \Phi_* \Lambda \Phi^\top [\Phi \Lambda \Phi^\top + I_N \otimes \sigma_r^2]^{-1} \Phi \Lambda \Phi_*^\top. \quad (4.36b)$$

However, since matrix dimensions are the same, it is not seen as a computational advantage yet. After algebraic manipulations such that:

$$\mathbf{m}^* = \mathbf{m}(\mathbf{x}^*) + \Phi_* \Lambda \Phi^\top [\Phi \Lambda \Phi^\top + I_N \otimes \sigma_r^2]^{-1} (\mathbf{z} - \mathbf{m}(\mathbf{x})), \quad (4.37a)$$

$$= \mathbf{m}(\mathbf{x}^*) + \Phi_* \Lambda [\Phi^\top \Phi \Lambda + I_N \otimes \sigma_r^2]^{-1} \Phi^\top (\mathbf{z} - \mathbf{m}(\mathbf{x})), \quad (4.37b)$$

$$= \mathbf{m}(\mathbf{x}^*) + \Phi_* [(\Phi^\top \Phi \Lambda + I_N \otimes \sigma_r^2) \Lambda^{-1}]^{-1} \Phi^\top (\mathbf{z} - \mathbf{m}(\mathbf{x})), \quad (4.37c)$$

$$= \mathbf{m}(\mathbf{x}^*) + \Phi_* [\Phi^\top \Phi \Lambda^{-1} + \Lambda^{-1} \otimes \sigma_r^2]^{-1} \Phi^\top (\mathbf{z} - \mathbf{m}(\mathbf{x})), \quad (4.37d)$$

$$= \mathbf{m}(\mathbf{x}^*) + \Phi_* [\Phi^\top \Phi + \Lambda^{-1} \otimes \sigma_r^2]^{-1} \Phi^\top (\mathbf{z} - \mathbf{m}(\mathbf{x})). \quad (4.37e)$$

Final forms are given below:

$$\mathbf{m}^* = \mathbf{m}(\mathbf{x}^*) + \Phi_* [\Phi^\top \Phi + \sigma_r^2 \Lambda^{-1}]^{-1} \Phi^\top (\mathbf{z} - \mathbf{m}(\mathbf{x})), \quad (4.38a)$$

$$\Sigma^* = \sigma_r^2 \Phi_* [\Phi^\top \Phi + \sigma_r^2 \Lambda^{-1}]^{-1} \Phi_*^\top. \quad (4.38b)$$

where usually prior means  $\mathbf{m}(\mathbf{x}^*)$  and  $\mathbf{m}(\mathbf{x})$  are often assumed to be 0 as mentioned before. Consequently, inversion of  $n \times n$  matrix as in equation 3.9, where  $n$  is the number of measurements, can now be reduced to the inversion of  $m \times m$  matrix where  $m$  is the number of bases. Moreover, it can now be decomposed in several forms, demonstrated in equation 4.37a to 4.37e. The use of equations 4.1 to 4.36 is summarized in Algorithm 4.1.

---

**Algorithm 4.1** Batch regression with reduced rank GP

---

- 1: Calculate  $m$  bases functions and  $\lambda$ 's using equation 4.33 or 4.17 concerning desired bounded domain limit  $L$  for each measurement
  - 2: Calculate  $\Phi_*$  for chosen test locations  $x_*$
  - 3: Construct  $\Phi$ ,  $\Phi_*$ , and  $\Lambda$  matrices to use in equation 4.35
  - 4: Calculate  $K(x, x^*)$ ,  $K(x^*, x)$ , and  $K(x^*, x^*)$  using equation 4.35
  - 5: Evaluate  $\mathbf{m}^*$  and  $\Sigma^*$  for predicted mean and covariance values with equation 4.36
- 

An example is given in Figure 4.4a for 16 measurements and 8 bases by implementing Algorithm 4.1 with Dirichlet boundary conditions in comparison with the full GP solution, which is introduced in equation 3.9. As can be seen in the figure, the reduced rank prediction is well suited to the full GP solution. The confidence interval is smaller in the interval from 0 to  $\frac{3\pi}{2}$  in which measurements are distributed. It is higher in the other part of the function. Since there is not a periodic pattern assumption in SE Kernel with Dirichlet boundary conditions, prediction values are different at the edges of the figure. Another example is given in Figure 4.5a for a periodic function with the same number of measurements and bases by using Algorithm 4.1 with periodic boundary conditions. It also testifies to consistency with the full GP solution. Moreover, since GP is customized concerning a periodic function in this example, although there are no measurements in the interval from  $\pi$  to  $2\pi$ , the function

is quite well predicted. It also shows GP's valuable advantage, which is considered as providing a confidence metric.

## 4.2 Recursive Form

Many applications require real-time solutions and cannot benefit from batch algorithms. Batch processing may be inadequate since one can only deduce after gathering all data. Therefore, online learning is becoming one of the essential requirements of a time dependent method. Particularly for tracking systems, a recursive processing method is a must-have. In this chapter, we will focus on making predictions with the reduced rank method by means of its recursive form.

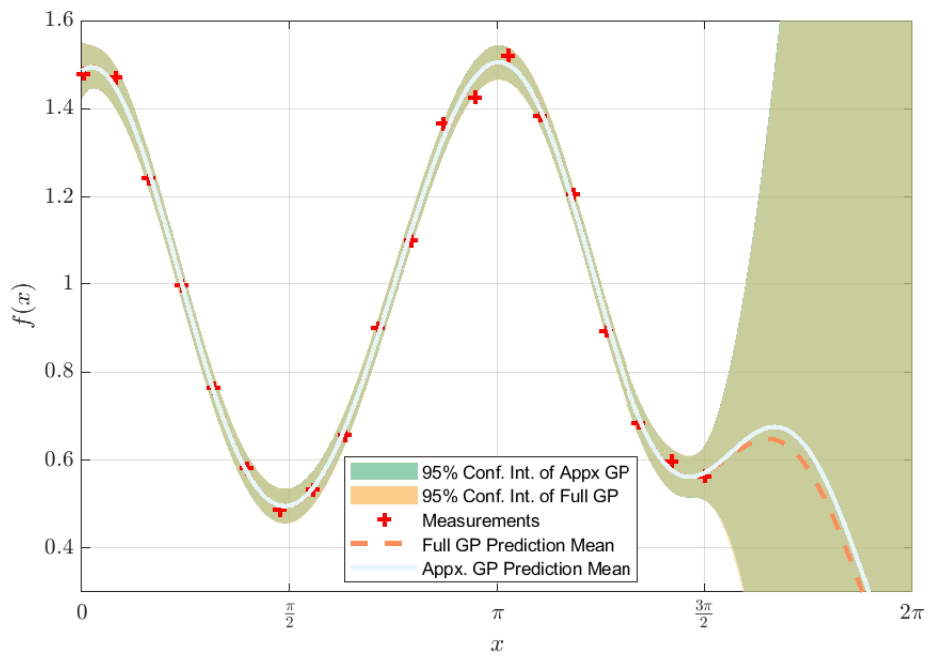
A closer look at equation 4.37a reveals that the desired mean value at test location  $x^*$  can be acquired with a matrix multiplication. For simplicity, the prior mean value for test location  $\mathbf{m}(x^*)$  is assumed to be 0. The posterior mean can be obtained just with a multiplication when needed such that:

$$\mathbf{m}^* = \Phi_* \boldsymbol{\mu}, \quad (4.39)$$

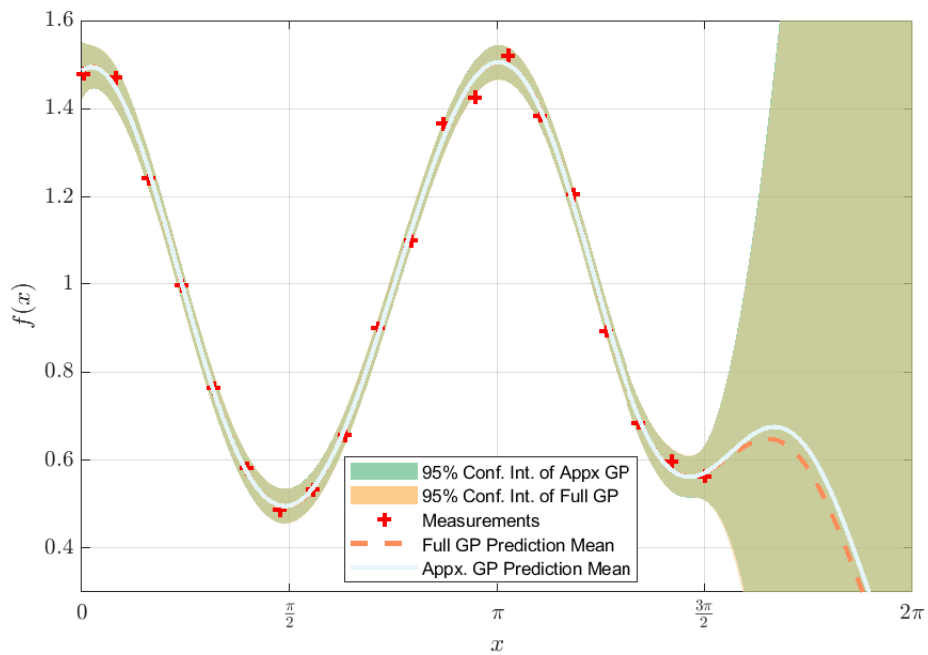
where  $\boldsymbol{\mu}$  is independent of the prediction point and can be written as follows

$$\boldsymbol{\mu} = \Lambda \Phi^\top [\Phi \Lambda \Phi^\top + I_N \otimes \sigma_r^2]^{-1} (\mathbf{z} - \mathbf{m}(\mathbf{x})). \quad (4.40)$$

The resulting expressions, which rely on conditional Gaussian densities, are similar to KF update equations. Initial mean and covariance can be assigned as  $\boldsymbol{\mu}_0 = 0$  and

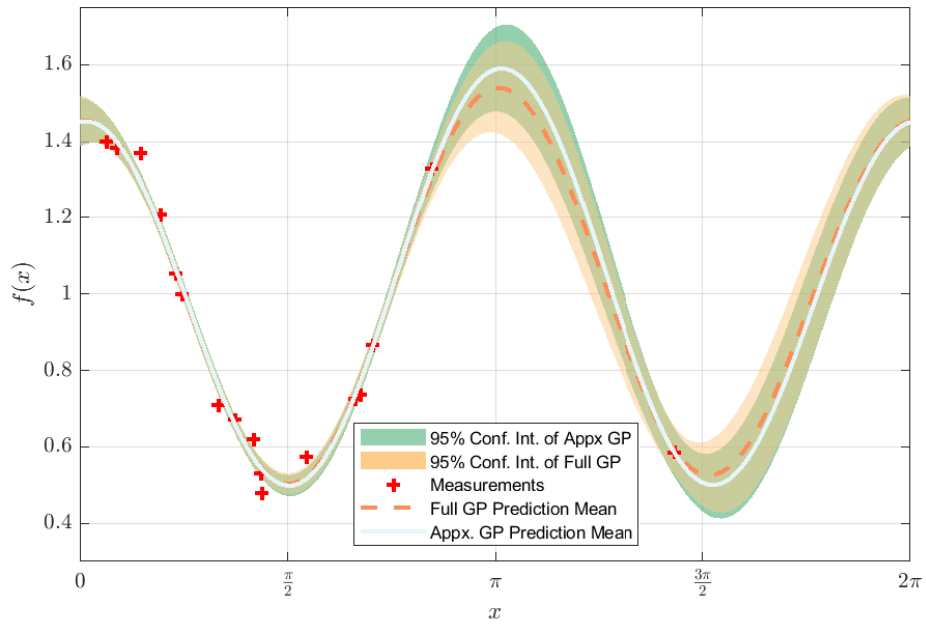


(a) Batch regression with reduced rank GP and full GP solution

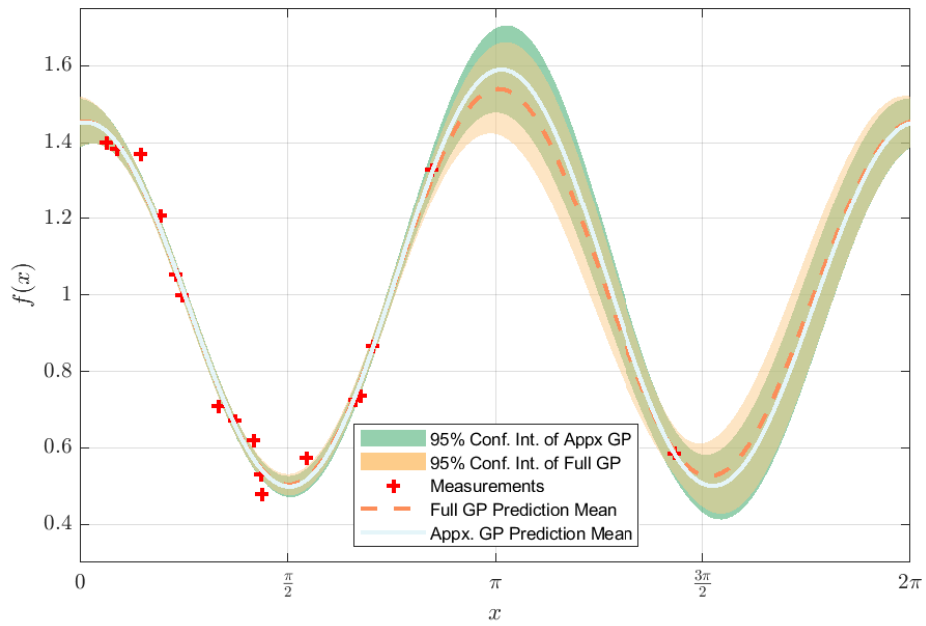


(b) The final result of recursive reduced rank GP regression and full GP solution

Figure 4.4: Prediction with reduced rank GP with Dirichlet boundary conditions and full GP SE kernel solution



(a) Batch regression with reduced rank GP and full GP solution



(b) The Final result of recursive reduced rank GP regression and full GP solution

Figure 4.5: Prediction with reduced rank GP with periodic boundary conditions and full GP periodic kernel solution

$\Sigma_0 = \Lambda$ . Subsequently, recursive forms can be expressed as:

$$\boldsymbol{\mu}_i = \boldsymbol{\mu}_{i-1} + K_i(\mathbf{y}_i - \Phi_i \boldsymbol{\mu}_{i-1}), \quad (4.41a)$$

$$\Sigma_i = \Sigma_{i-1} - K_i \Phi_i \Sigma_{i-1}, \quad (4.41b)$$

$$K_i = \Sigma_{i-1} \Phi_i^T S_i^{-1}, \quad (4.41c)$$

$$S_i = \Phi_i \Sigma_{i-1} \Phi_i^T + \sigma_r^2 I. \quad (4.41d)$$

Furthermore, the covariance form given in equation 4.41b can be rearranged by substituting the defined Kalman gain  $K$  and innovation matrix  $S$  given in equations 4.41c and 4.41d, respectively, into the form  $K_i S_i K_i^T$  such that

$$K_i S_i K_i^T = \Sigma_{i-1} \Phi_i^T (\Phi_i \Sigma_{i-1} \Phi_i^T + \sigma_r^2 I)^{-1} (\Phi_i \Sigma_{i-1} \Phi_i^T + \sigma_r^2 I) K_i^T, \quad (4.42a)$$

$$= \Sigma_{i-1} \Phi_i^T K_i^T, \quad (4.42b)$$

$$= (K_i \Phi_i \Sigma_{i-1}^T)^T. \quad (4.42c)$$

Since covariance matrices must be symmetric, the last term can be written as  $K_i \Phi_i \Sigma_{i-1}$ , which is exactly the same term in equation 4.41b or equation 2.9d. Moreover, the covariance measurement update equation becomes

$$\Sigma_i = \Sigma_{i-1} - K_i S_i K_i^T. \quad (4.43)$$

Consequently, the state-space model of this recursive learning model can be written for the state  $\mathbf{x}$  such that

$$\mathbf{x}_{i+1} = \mathbf{x}_i, \quad (4.44a)$$

$$y_i = \Phi_i \mathbf{x}_i + e_k, \quad e_k \sim \mathcal{N}(0, R) \quad (4.44b)$$

where  $e_k$  is the measurement noise with covariance matrix  $R$ , and  $y_i$  is the measurement at time  $i$ .

There is no need to evaluate any variable with regard to test inputs in this formulation. When we need the predicted mean and covariance values at a test location, it can be calculated as

$$\mathbf{m}_i^* = \Phi_* \boldsymbol{\mu}_i, \quad (4.45a)$$

$$\Sigma_i^* = \Phi_* \Sigma_i \Phi_*^T. \quad (4.45b)$$

The procedure for this recursive estimation is summarized in Algorithm 4.2. After obtaining all measurements, there is no difference between batch and recursive predictions since they lie at the root of the same equation regardless of the kernel function. The same functions given in Figure 4.4a and Figure 4.5a are predicted with Algorithm 4.2. The final results are given in Figure 4.4b and Figure 4.5b. As shown in the figures, there is no difference between the two implementations for each function. This formulation may also be beneficial in other applications since it can be used with augmented structures introduced in Section 5 for the ETT problem.

---

**Algorithm 4.2** Recursive regression with reduced rank GP

---

- 1: Initialize  $\boldsymbol{\mu}_0 = 0$  and  $\Sigma_0 = \Lambda$  where  $\Lambda$  is introduced in equation 4.34
  - 2: **for**  $i = 1, 2, \dots, n$  **do**
  - 3:     Calculate  $\phi_i$  with measurement location  $x_i$
  - 4:     Construct  $\Phi_i$  as in equation 4.34
  - 5:     Calculate  $\boldsymbol{\mu}_i$  and  $\Sigma_i$  with equations 4.41a to 4.41d
  - 6:     When a prediction is needed
  - 7:     Construct  $\Phi_*$  for desired prediction locations  $\mathbf{x}^*$
  - 8:     Evaluate  $\mathbf{m}_i^*$  and  $\Sigma_i^*$  by equation 4.45a and 4.45b
  - 9: **end for**
-

### 4.3 Learning Target Extent Using Reduced Rank Gaussian Process Regression

The instruments adopted in the field of machine learning, or more comprehensively artificial intelligence, are features. They are undergoing a revolution in terms of engineering, science, or applied mathematics and becoming the key component of algorithms [17]. The target extent is a remarkable feature to investigate. It defines the shape of the object, which is valuable information to assist researchers in exploring targets. Preliminary works are conducted for modeling the extent as circle, rectangle, or ellipsoids [11], [24], [38]. Recently, modeling the target extent as star-convex shapes is generating considerable interest in terms of flexibility, attractive formulation, and match with real objects. The mathematical definition can be asserted such that if there exists a point  $\zeta_0$  in a set  $S \in R^n$  where any line segment from  $\zeta_0$  to any other point  $\zeta$  contained in  $S$ ,  $S$  becomes a star domain. Figure 4.6 shows an arbitrary star-convex shape with regard to terminology given in the previous definition. Moreover, convex is a more general property than star-convex, i.e., every convex set is also star-convex, but not vice versa.

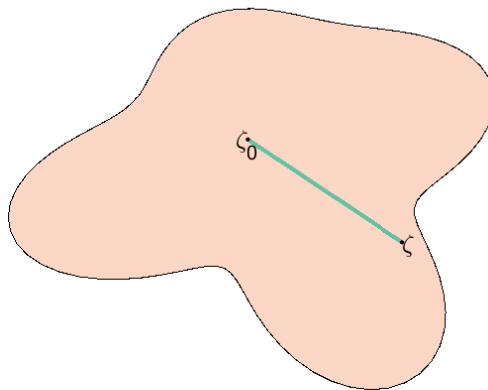


Figure 4.6: An arbitrary star domain with an example line segment from  $\zeta_0$  to  $\zeta$

Various approaches are carried out by means of star-convex modeling of shapes for

tracking purposes [10], [63], [50]. The 2D shape, which is modeled as expressed previously, can also be represented in the polar domain, referred to as  $r - \theta$  interchangeably. Hence, Gaussian processes can be utilized to learn the target extent with measurements arising on the contours. The first systematic study to learn the object's boundary with GP is suggested in [66]. Moreover, [41] proposed an enhancement of it to 3 dimensional space. Furthermore, by learning the target extent, classification or recognition can also be performed [65]. Different objects and their representation in  $r - \theta$  domain are depicted in Figure 4.7 and Figure 4.8. The polar domain representations can be declared as in Table 4.1 for different types of shapes with a simple equation.

Table 4.1: Equations for radius with respect to the angle which starts from the center of the shape [15]

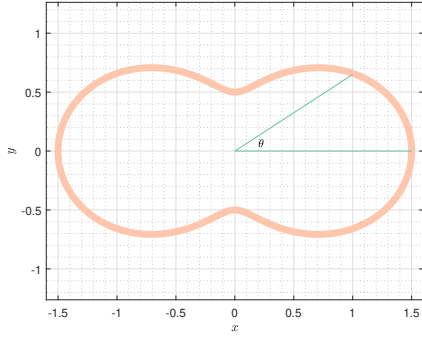
Shape	Equation
Peanut	$r = 1 + \frac{1}{2} \sin(2\theta + \frac{\pi}{2})$
Square	$r = \min \left( \frac{1}{ \cos(\theta) }, \frac{1}{ \sin(\theta) } \right)$
Equilateral Triangle	$r = \frac{\cos(\frac{\pi}{3})}{\cos((\text{mod}(\theta, \frac{2\pi}{3})) - \frac{\pi}{3})}$
Regular n-gon	$r = \frac{\cos(\frac{\pi}{n})}{\cos((\text{mod}(\theta, \frac{2\pi}{n})) - \frac{\pi}{n})}$

Further, if we choose  $\zeta_0$  as the center of the object, we can model its boundary, measuring the length from the center  $r$  with respect to  $\theta$ . Hence,  $r = f(\theta)$  can be hypothesized as

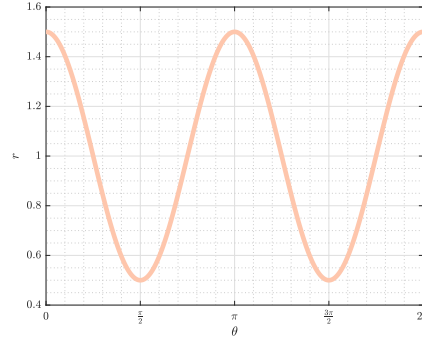
$$f(\theta) = \mathcal{GP}(0, K), \quad (4.46)$$

where  $K$  is the kernel function introduced previously. In order to recover the whole shape, it is defined in interval  $[0, 2\pi]$  for this approach.

An example shape is chosen to illustrate the learning process. A stationary peanut-like shape is learned with a total of 9 measurements recursively by implementing reduced rank GP as proposed in Algorithm 4.2. The number of bases is chosen 16,

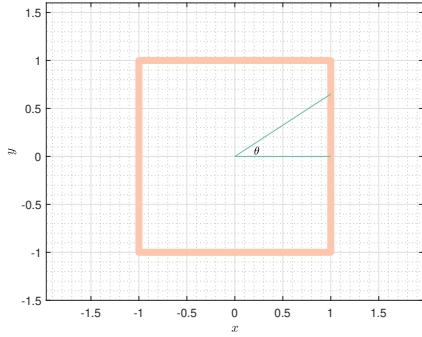


(a) Target extent in Cartesian representation

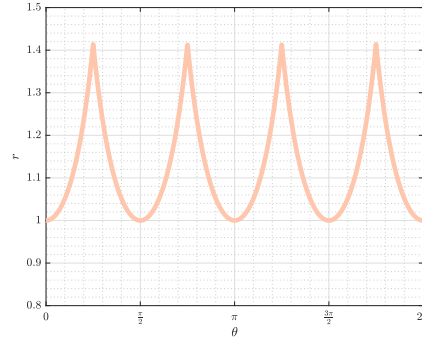


(b) Target extent in  $r - \theta$  representation

Figure 4.7: An illustration of target extent representations for a peanut-like shape in different domains



(a) Target extent in Cartesian representation



(b) Target extent in  $r - \theta$  representation

Figure 4.8: An illustration of target extent representations for a square shape in different domains

and they are calculated for the squared exponential kernel given in Subsection 4.1.1. The prediction is evaluated at four different time instances. Results belonging to four different steps of recursion are given in Figure 4.9. Red plus signs indicate measurements, and the dashed line shows the true shape for each time step. The blue line illustrates the mean value of the predicted shape. As can be seen in example time instances, areas which there are no measurements indicated with high uncertainty levels. The uncertainty level is lower near measurement locations.

The final result is enlarged in Figure 4.10. The predicted extent substantially coincides with the true extent. However, a discontinuity problem occurs near the angles 0 and  $2\pi$ . This problem indicates that the information can not be conveyed through

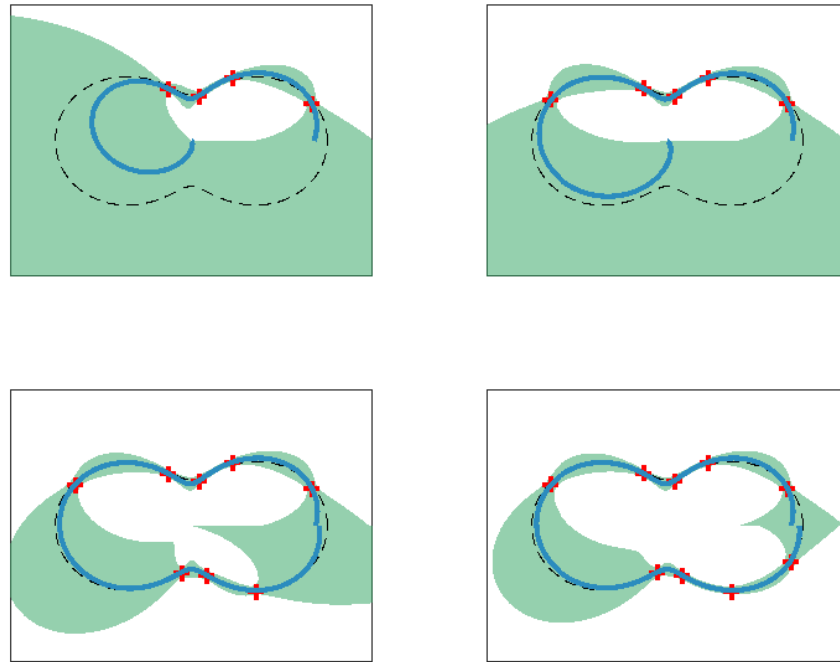


Figure 4.9: Recursive reduced rank GP prediction of a peanut-like shape with the squared exponential kernel in different time instances

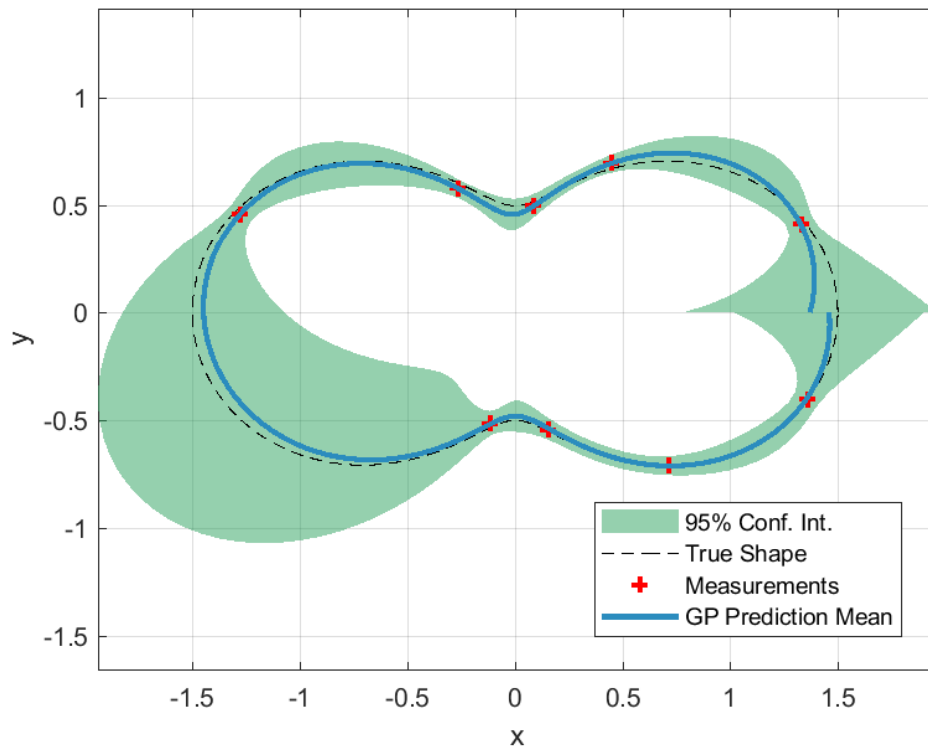


Figure 4.10: Recursive reduced rank GP prediction of a peanut-like shape with the squared exponential kernel

the edges. For this reason, the periodic kernel, a solution to this problem, is utilized. The same example is learned with the periodic kernel, and the final result is given in Figure 4.11. The resulting predicted extent is continuous, and the problem confronted with the SE kernel is resolved.

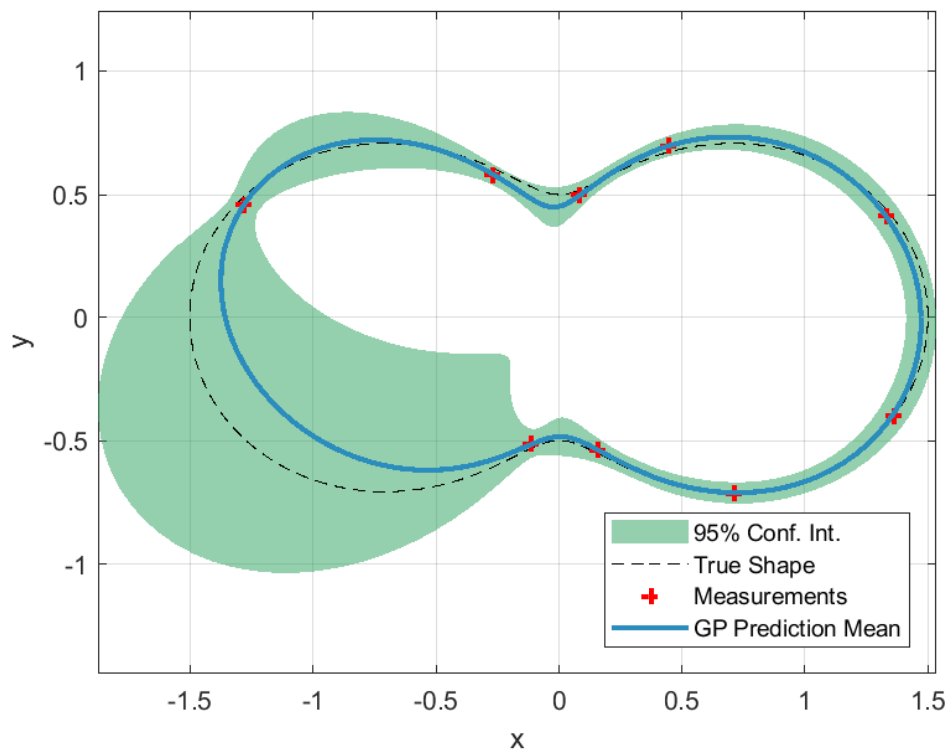


Figure 4.11: Recursive reduced rank GP prediction of a peanut-like shape with the periodic kernel



## CHAPTER 5

### AUGMENTED STATE SPACE MODEL FOR EXTENDED TARGET TRACKING

Standard approaches in target tracking consider targets as point sources. In the history of target tracking, the focus has exclusively been given to estimating the target's kinematics such as location, velocity, or acceleration [44], [49]. With the advances in sensor technology, it is now possible to extract more information from available measurements and estimate target extent simultaneously with the target kinematic states [66]. Consider the augmented state vector for time  $k$

$$\mathbf{x}_k \triangleq \left[ \bar{\mathbf{x}}_k^\top \quad (\mathbf{x}_k^s)^\top \right]^\top, \quad (5.1)$$

where  $\mathbf{x}_k^s$  is the target extent related state variables and  $\bar{\mathbf{x}}_k$  consists of the target location, orientation, and velocities. They are denoted as  $\mathbf{x}_k^c$ ,  $\psi_k$ , and  $\mathbf{x}_k^v$  respectively and can be written in the form

$$\bar{\mathbf{x}}_k \triangleq \left[ (\mathbf{x}_k^c)^\top \quad \psi_k \quad (\mathbf{x}_k^v)^\top \right]^\top. \quad (5.2)$$

In order to associate the kinematic state variables and the target extent related variables in one vector, two different coordinate systems are described. The first one is the global coordinate frame, where measurements and targets are observed within a space, including a sensor. The other one is termed as body frame, which might be revolved in accordance with the target with respect to the global coordinate system with an angle  $\psi$ . Two different coordinate system representations are given in Figure 5.1

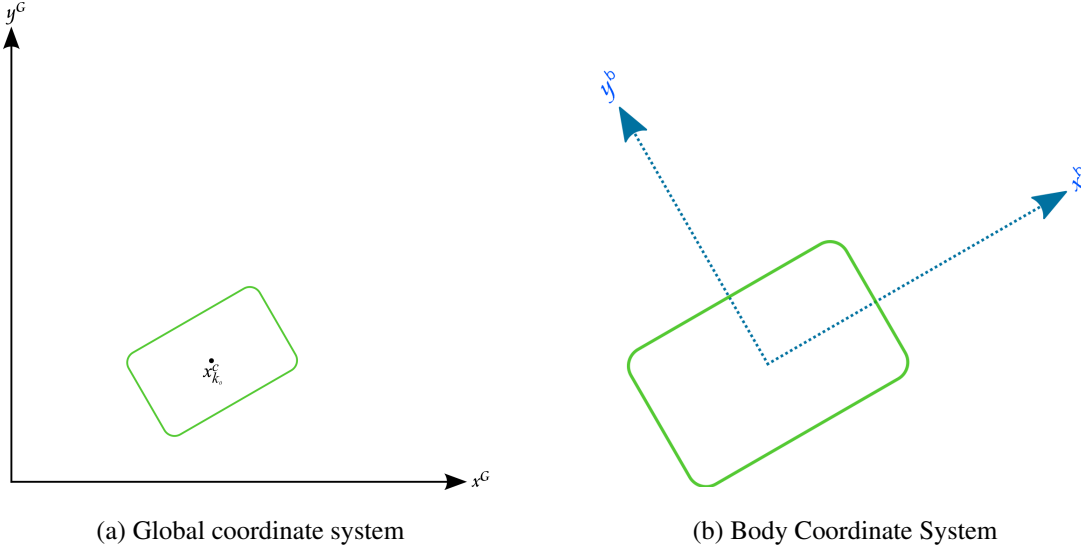


Figure 5.1: Coordinate systems representations of an extended target

for an example extended target.  $(\cdot)^G$  stands for a formulation in the global coordinate frame and  $(\cdot)^b$  for body coordinates.

An angle in the global coordinate frame with respect to target center  $\mathbf{x}_k^c$  at time  $k$  can be defined as

$$\theta_{k,l}^G(\mathbf{x}_k^c) = \angle(\mathbf{y}_{k,l} - \mathbf{x}_k^c), \quad (5.3)$$

where  $\mathbf{y}_{k,l}$  is the  $l^{\text{th}}$  measurement at time  $k$ . Coordinate transformation can be applied with the following equation, yet, more comprehensive work has been carried out in [41] for higher dimensional space using quaternions

$$\theta_{k,l}^b(\mathbf{x}_k^c, \psi_k) = \theta_{k,l}^G(\mathbf{x}_k^c) - \psi_k. \quad (5.4)$$

These variables are shown in Figure 5.2, with target shapes belonging to consecutive time instances at the same graph.

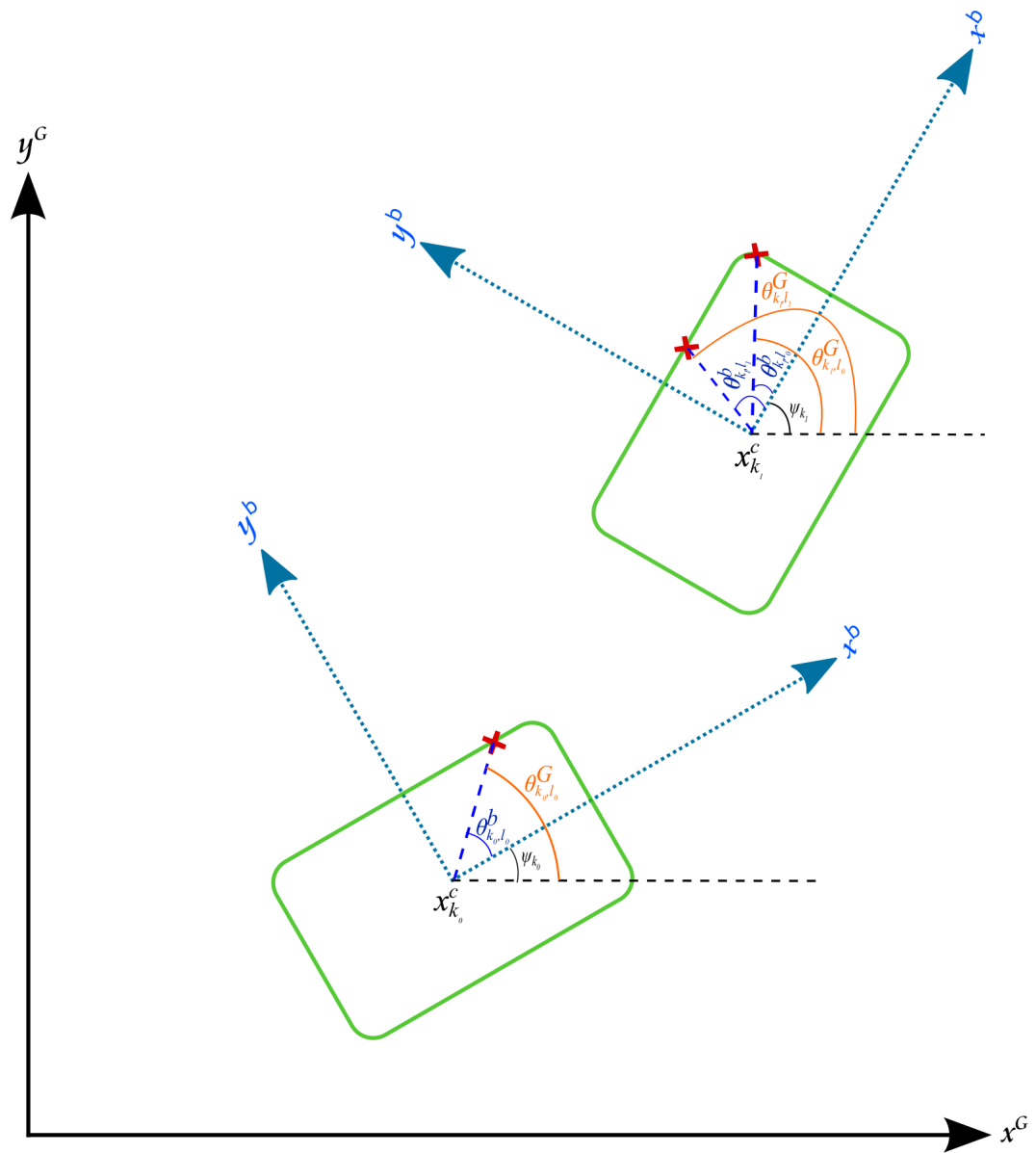


Figure 5.2: Different coordinate systems of the same extended target at two consecutive time instances as  $k_0$  and  $k_1$ . Red cross signs represent the measurements as  $l_0$  and  $l_1$  belonging to each time

After acquiring angle information in the body frame, the underlying function that corresponds to the target extent at that location, which is modeled as  $\mathcal{GP}$  as in equation 4.46, can be obtained. It is plausible that the measurements are influenced not only by the target extent but also by location, coordinate transformation, and noise as well. Therefore,  $l^{\text{th}}$  measurement at time instance  $k$ , which is represented as  $\mathbf{y}_{k,l}$  and comprises two elements, can be modeled by transforming  $f(\theta_{k,l}^b(\mathbf{x}_k^c, \psi_k))$  to the Cartesian frame by multiplying with  $\mathbf{p}(\theta_{k,l}^G(\mathbf{x}_k^c))$ , then moving its center to the global coordinate frame by adding  $\mathbf{x}_k^c$ , subsequently adding the noise. This sequence of operations can be formulated as

$$\mathbf{y}_{k,l} = \mathbf{x}_k^c + \mathbf{p}(\theta_{k,l}^G(\mathbf{x}_k^c)) f(\theta_{k,l}^b(\mathbf{x}_k^c, \psi_k)) + \dot{\mathbf{e}}_{k,l}, \quad (5.5)$$

where  $\mathbf{p}(\theta_{k,l}^G(\mathbf{x}_k^c))$  is the orientation vector and written as  $\mathbf{p}_{k,l}(\mathbf{x}_k^c)$  for simplicity and can be expressed as

$$\mathbf{p}(\theta_{k,l}^G(\mathbf{x}_k^c)) = \mathbf{p}_{k,l}(\mathbf{x}_k^c) = \frac{\mathbf{y}_{k,l} - \mathbf{x}_k^c}{\|\mathbf{y}_{k,l} - \mathbf{x}_k^c\|}. \quad (5.6)$$

An additional substitution can be made for  $f(\cdot)$  by replacing it with the previous derivation as in equation 4.44b. The resulting equation can be simplified as

$$\mathbf{y}_{k,l} = \mathbf{x}_k^c + \mathbf{p}_{k,l}(\mathbf{x}_k^c) [\Phi(\theta_{k,l}^b(\mathbf{x}_k^c, \psi_k)) \mathbf{x}_k^s + \mathbf{e}_{k,l}^s] + \dot{\mathbf{e}}_{k,l}, \quad (5.7a)$$

$$= \underbrace{\mathbf{x}_k^c + \tilde{H}_l(\mathbf{x}_k^c, \psi_k) \mathbf{x}_k^s}_{=\mathbf{h}_{k,l}(\mathbf{x}_k)} + \underbrace{\mathbf{p}_{k,l}(\mathbf{x}_k^c) \mathbf{e}_{k,l}^s + \dot{\mathbf{e}}_{k,l}}_{=\mathbf{e}_{k,l}}, \quad (5.7b)$$

$$= \mathbf{h}_{k,l}(\mathbf{x}_k) + \mathbf{e}_{k,l}. \quad (5.7c)$$

New measurement noise can be defined as a Gaussian density with covariance  $R_{k,l}$  such that

$$\mathbf{e}_{k,l} \sim \mathcal{N}(0, R_{k,l}), \quad (5.8)$$

where  $k$  shows the time, and  $l$  represents the measurement number. Other works, as an example [66], use fixed and predetermined points on the extent as state variables. This requires a transformation operation to associate measurements with points in the state concerning their locations. Therefore, they calculate  $R_{k,l}$  for each measurement due to their modeling approach. However, in contrast with what was previously thought and proposed in other works, the reduced rank GP method does not necessitate these calculations. Measurement noise covariance values can be taken the same and predetermined in the reduced rank GP method with a variance  $\sigma_r^2$  such that

$$R_{k,l} = \sigma_r^2 \otimes I. \quad (5.9)$$

Moreover,  $\mathbf{h}_{k,l}$  in equation 5.7c can be rewritten as follows

$$\mathbf{h}_{k,l}(\mathbf{x}_k) = \mathbf{x}_k^c + \tilde{H}_l(\mathbf{x}_k^c, \psi_k) \mathbf{x}_k^s, \quad (5.10a)$$

$$\tilde{H}_l(\mathbf{x}_k^c, \psi_k) = \mathbf{p}_{k,l}(\mathbf{x}_k^c) \Phi(\theta_{k,l}^b(\mathbf{x}_k^c, \psi_k)). \quad (5.10b)$$

The first set of formulations have highlighted the impact of measurements on the variables. Besides, the kinematic state transition can be written similar to equation 2.6a as follows

$$\bar{\mathbf{x}}_{k+1} = \bar{F}\bar{\mathbf{x}}_k + \bar{\mathbf{w}}_k, \quad (5.11)$$

where the process noise for the kinematic state can be expressed as zero-mean Gaussian with covariance matrix  $\bar{Q}$  such that

$$\bar{\mathbf{w}}_k \sim \mathcal{N}(0, \bar{Q}). \quad (5.12)$$

$\bar{F}$  and  $\bar{Q}$  can be defined with the following two equations for the constant velocity

model with sampling time  $T$

$$\bar{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes I_3 \quad (5.13a)$$

$$\bar{Q} = \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \otimes \begin{bmatrix} \sigma_c^2 & 0 & 0 \\ 0 & \sigma_c^2 & 0 \\ 0 & 0 & \sigma_\psi^2 \end{bmatrix}. \quad (5.13b)$$

$\sigma_c^2$  and  $\sigma_\psi^2$  correspond for the process noise variances for position and orientation, respectively.

By combining equation 5.11 with the previous extent related state, the augmented state transition equation can be written such that

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{w}_k, \quad (5.14a)$$

$$\mathbf{w}_k \sim \mathcal{N}(0, Q). \quad (5.14b)$$

These tailored structures can be constructed in an augmented matrix forms where  $(\cdot)^s$  represents extent related variables,  $(\cdot)$  corresponds to motion-related variables as follows

$$\mathbf{x}_k = \begin{bmatrix} \bar{\mathbf{x}}_k \\ \mathbf{x}_k^s \end{bmatrix}, \quad (5.15a)$$

$$F = \begin{bmatrix} \bar{F} & 0 \\ 0 & I_m \end{bmatrix}, \quad (5.15b)$$

$$Q = \begin{bmatrix} \bar{Q} & 0 \\ 0 & 0_m \end{bmatrix}, \quad (5.15c)$$

where  $m$  is the number of bases and  $0_m$  represents  $m \times m$  zero matrix.

Complementary attempts are accomplished with the purpose of using all measurements in one time instance in [66] and in [41]. So, each variable can be concatenated to involve entire measurements with the following augmented variables by using equation

$$\mathbf{y}_k = \left[ y_{k,1}, \quad \dots, \quad y_{k,n_k} \right]^\top, \quad (5.16a)$$

$$R_k = R_{k,l} \otimes I_{n_k}, \quad (5.16b)$$

$$\mathbf{h}_k(\mathbf{x}_k) = \left[ \mathbf{h}_{k,1}(\mathbf{x}_k)^\top, \quad \dots, \quad \mathbf{h}_{k,n_k}(\mathbf{x}_k)^\top \right]^\top. \quad (5.16c)$$

In the light of recent derivations, a simple, compact form can be written to utilize filtering operations as

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{w}_k, \quad (5.17a)$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{e}_k, \quad (5.17b)$$

where  $\mathbf{e}_k \sim \mathcal{N}(0, R_k)$ , with covariance  $R_k$  written in equation 5.16b, and  $\mathbf{w}_k$  is given in 5.14b. Even though initially extent related state and measurements are defined in a linear form, augmenting motion variables relate them nonlinearly with  $\mathbf{h}_k$ . This inevitably leads to other variants of KF, for instance, EKF introduced in Chapter 2.1, which is implemented for this work as well as for [66], [41], or [65]. The resulting algorithm is given in Algorithm 5.1.

## 5.1 Proof of the Concept

The proposed algorithm is applied to an example scenario. A peanut-like shape target is considered for a period of 30 time instances where 8 measurements are observed per time scan with a uniform distribution. The target follows a path that includes a turn

---

**Algorithm 5.1** Extended Target Tracking with Reduced Rank Gaussian Processes

---

- 1: Initialize  $\mu_0$  and  $P_0$ ,
  - 2: Construct  $\Phi_*$  for desired evaluation locations  $\mathbf{x}^*$ ,
  - 3: **for** Consecutive time instances  $k = 1, 2, \dots$  **do**
  - 4:     Make EKF predictions in new time instance  $k$ ,
  - 5:     **for** Measurements  $l = 1, 2, \dots, n_k$  **do**
  - 6:         Obtain the angle in global coordinate frame  $\theta_{k,l}^G$  by using  $\mathbf{y}_{k,l}$  with  $\mathbf{x}_k^c$ ,
  - 7:         Calculate the body angle  $\theta_{k,l}^b$  as in equation 5.4 with current  $\psi_k$ ,
  - 8:         Construct  $\bar{\Phi}(\theta_{k,l}^b(\mathbf{x}_k^c, \psi_k))$  by analytically derived bases with eq. 4.33,
  - 9:         Determine the orientation vector  $\mathbf{p}_{k,l}$  with  $\mathbf{x}_k^c$  as in eq. 5.6,
  - 10:         Calculate  $\tilde{H}(\cdot)$  and  $\mathbf{h}_{k,l}(\cdot)$  with equation 5.10a and 5.10b,
  - 11:     **end for**
  - 12:     Construct the structures  $y_k$ ,  $R_k$ , and  $\mathbf{h}_k$  expressed in equation 5.16a to 5.16c,
  - 13:     Make EKF measurement update with obtained variables,
  - 14:     If an evaluation is demanded for the extent, multiply  $\Phi_*$  with  $\mathbf{x}_k^s$  as in eq. 4.45a and use equation 4.45b for covariance.
  - 15: **end for**
-

after a while and continues its motion. The constant velocity model is implemented to utilize EKF equations for this demonstration [13]. The number of bases is defined as 9 for the target extent, and structures are arranged as described in Algorithm 5.1.

Figure 5.3 illustrates a visualization of the whole path with sampled time instances and highlights two specific drawings belonging near initial and last evaluations. As can be seen in Figure 5.3 (a) target center, orientation, and its extent are tracked simultaneously with considerable success. In Figure 5.3 (b), it can be seen that although the target is at the path's beginning, the ETT result is substantially acceptable. It also reveals the uncertain parts of the target for which measurements are not observed yet. Figure 5.3 (c) shows the result of the estimation at the end of the path, with red plus signs for measurements. It is an enlarged version of the final step to assert a prosperous track where the dashed peanut-like shape resembles the true extent. The true extent and estimation radically overlap in addition to the target center and orientation.

Estimations and ground truths are depicted in Figure 5.4 with 5.4a and 5.4b graphs. The former shows true positions with dashed lines for 2 dimensions, and the latter reveals angle estimation in comparison with ground truth, which is coherently drawn as a dashed line. As shown in Figure 5.4, visual overlapping in Figure 5.3 can be seen with a different aspect.

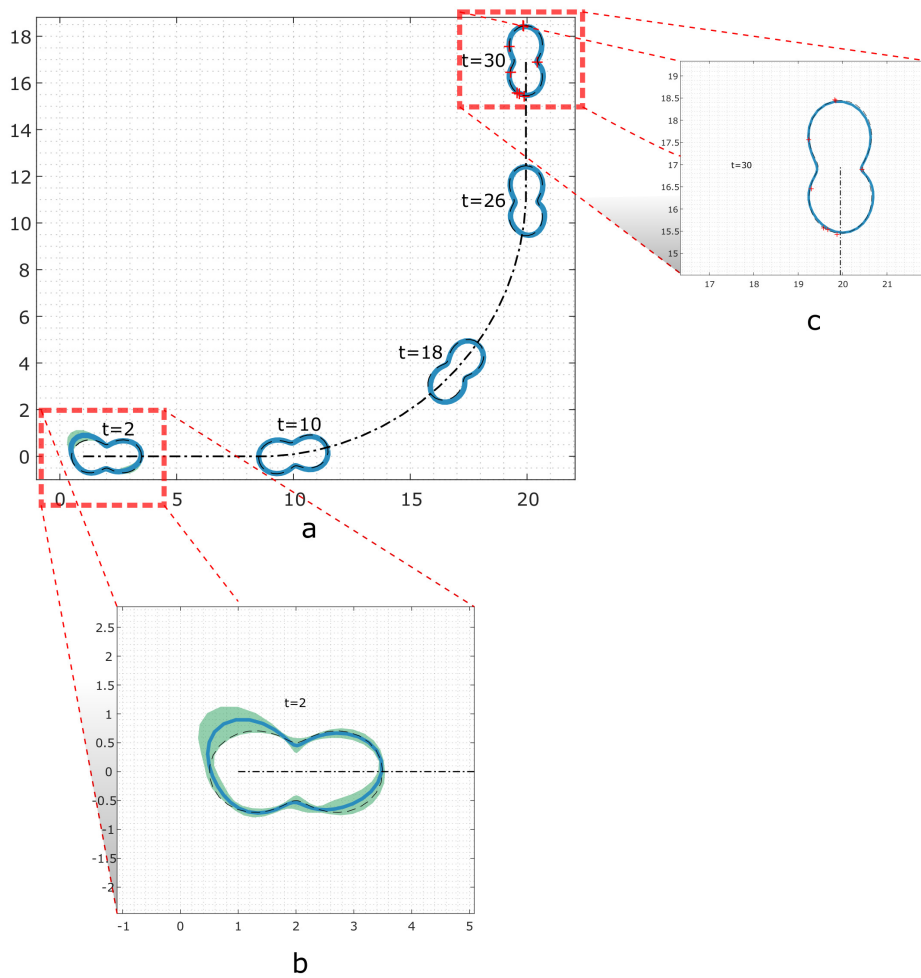
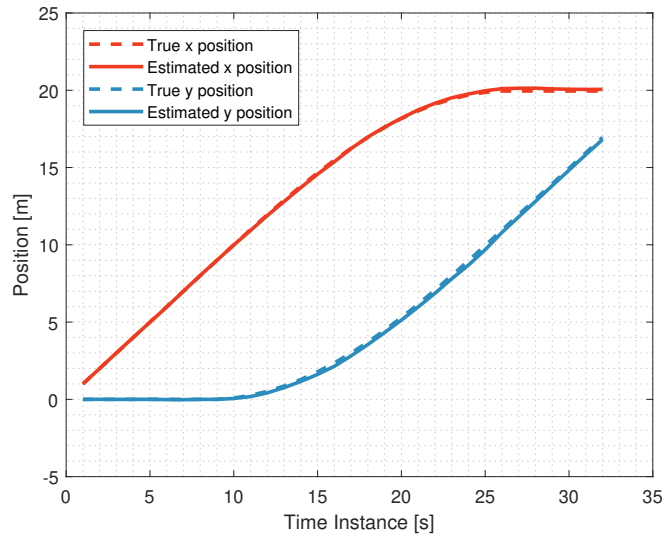
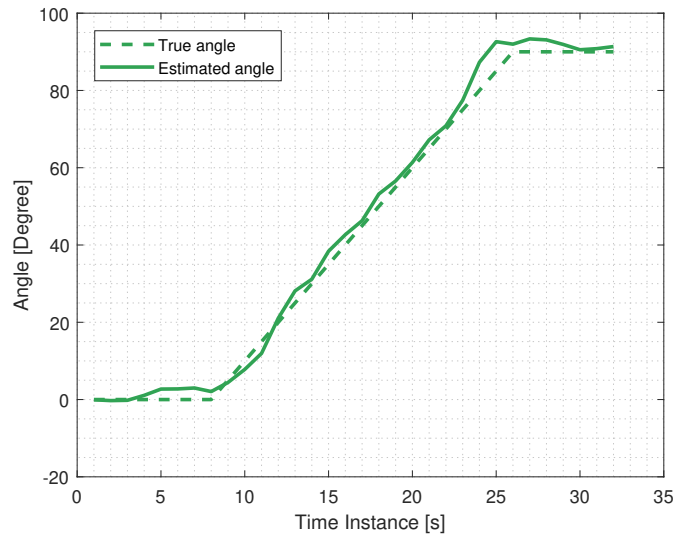


Figure 5.3: An example ETT result of a peanut-like shape by using Algorithm 5.1 (a) General view with a few sampled time instances, (b) Enlarged result of near initial time, (c) Magnified final time instance



(a) Position estimations and true positions



(b) Orientation estimation and true direction

Figure 5.4: Estimation and ground truth graphics for positions and orientation for visualization in Figure 5.3

## 5.2 Computational Aspects

Computational complexity is a measure that shows how fast an algorithm can work. It defines computational demands that methods require to make calculations. We especially pointed out the time complexity, which quantifies the amount of time that an algorithm entails to complete its routine, instead of memory requirements. In this section, we examine the computational differences analytically in terms of the total number of operations where  $\mathcal{O}(\cdot)$  notation is used to express the time complexity as a figure of merit.

Other works in the field of ETT with GPs utilize a different approximation by storing physical quantities in the state vector [66, 41, 65]. The values in the state correspond to the radiuses for predetermined angles. On the other hand, the proposed method only stores coefficients for bases, which can be calculated by an analytical function. Therefore, the two approaches principally differ in the way of the nonlinear observation model affecting computational aspects.

The measurement update steps between the method in [66], denoted as ETT-GP, and the proposed method ETT using reduced rank Gaussian processes (ETT-RRGP) are compared in table 5.1.

Table 5.1: Comparison table for measurement update steps in ETT-GP and ETT-RRGP

ETT-GP Measurement Update Steps	ETT-RRGP Measurement Update Steps
Calculate $H^f$	Construct $\Phi(\cdot)$ matrix
Calculate $R^f$	Use constant $R$ matrix
Compute Jacobians	Compute Jacobians
Perform KF Measurement Update	Perform KF Measurement Update

The computational requirements of the last step of both updates are the same if the algorithms share a common number of basis vectors. However, ETT-RRGP algorithm can perform well with a fewer number of basis compared to ETT-GP. The differences

in the computation of the Jacobians are minor. The significant components which create the differences in the execution time of the algorithms are the computation of  $H^f$  and  $R^f$  matrices in ETT-GP, and the construction of the  $\Phi(\cdot)$  matrix in ETT-RRGP. We will investigate the complexity of these steps in more detail. The number of bases is referred as  $m$ , and the number of measurements as  $n$ .

- $H^f$  matrix computation:

The observation model matrix proposed for ETT-GP can be written as

$$H^f(\mathbf{u}) = K(\mathbf{u}, \mathbf{u}^f)(K(\mathbf{u}^f, \mathbf{u}^f))^{-1}, \quad (5.18)$$

where  $\mathbf{u}$  is the vector for measurement locations and  $\mathbf{u}^f$  for basis point locations.

Matrix inverse costs  $\mathcal{O}(m^3)$  operations but can be compensated by storing in the memory because basis point locations are not changed over time. Even though  $(K(\mathbf{u}^f, \mathbf{u}^f))^{-1}$  can be acquired offline and stored, there is still a need for multiplication. Multiplication performs  $\mathcal{O}(nm^2)$  operations for this equation.

- $R^f$  matrix computation:

In the method proposed by [66], the state vector consists of radiuses for specific angles. However, measurements are not obtained exactly in these angles. Therefore, they have to perform a measurement noise covariance matrix calculation routine by the following variable for equation 5.9

$$R^f(\mathbf{u}) = K(\mathbf{u}, \mathbf{u}) + R - K(\mathbf{u}, \mathbf{u}^f) [K(\mathbf{u}^f, \mathbf{u}^f)]^{-1} K(\mathbf{u}^f, \mathbf{u}), \quad (5.19)$$

where  $\mathbf{u}^f$  stands for basis point locations,  $\mathbf{u}$  is the input vector for measurement locations,  $R$  is a constant diagonal deviation matrix, and  $K(\cdot)$  is the kernel's matrix form. Significant operations in Equation 5.19 requires  $\mathcal{O}(nm^2)$  time complexity in total.

- $\Phi(\cdot)$  construction:

$\Phi$  matrix is constructed for  $m$  bases with the input vector for measurement locations written as  $\mathbf{u} = [u_1, u_2, \dots, u_n]$  such that

$$\Phi = \begin{bmatrix} \phi_1(u_1) & \dots & \phi_m(u_1) \\ \vdots & \ddots & \vdots \\ \phi_1(u_n) & \dots & \phi_m(u_n) \end{bmatrix}, \quad (5.20)$$

where  $\phi(\cdot)$  is a  $\cos(\cdot)$  or a  $\sin(\cdot)$  value that analytical formulas are given in equation 4.33. Since there is neither a matrix multiplication nor a matrix addition, this construction's time complexity only consists of function evaluations.

The summary of the significant time complexities between the two methods is comparably presented in table 5.2. To make numeric comparisons, typical scenarios are investigated with total time complexities for the major differences in table 5.3.

Table 5.2: Significant time complexities in ETT-GP for different variables compared to ETT-RRGP.  $\mathcal{O}(\cdot)^*$  shows the complexity due to matrix inverse operations, which can be dismissed by precalculations and storing in memory. The number of bases is denoted as  $m$ , and  $n$  refers to the number of measurements.

ETT-GP Step	Time Complexity	ETT-RRGP Equivalent	Time Complexity
$H^f(u)$	$\mathcal{O}(nm^2) + \mathcal{O}(nm^3)^*$	$\Phi(\cdot)$	$\mathcal{O}(1)$
$R^f(u)$	$\mathcal{O}(nm^2) + \mathcal{O}(nm^3)^*$	$R$	$\mathcal{O}(0)$

As indicated previously, the complexity of matrix inverse operations can be handled by storing them in memory. Consequently, the analytical differences in time complexity can be declared as quadratically increasing with the number of basis points, and it is linearly increasing with the number of measurements by using naive matrix multiplication algorithms.

In addition to the significant components which create the differences in the execution time of the algorithms, the time complexities of common steps, both of which exploit

Table 5.3: Number of operations concerning significant terms for typical 2-D and 3-D scenarios in comparison with ETT-GP and ETT-RRGP

Method	Typical 2-D Scenario with 50 bases and 10 measurements	Typical 3-D Scenario with 1000 bases and 10 measurements
ETT-GP	25000	10000000
ETT-RRGP	500	10000

the same EKF update equations introduced in equation 2.12 are also presented in table 5.4.

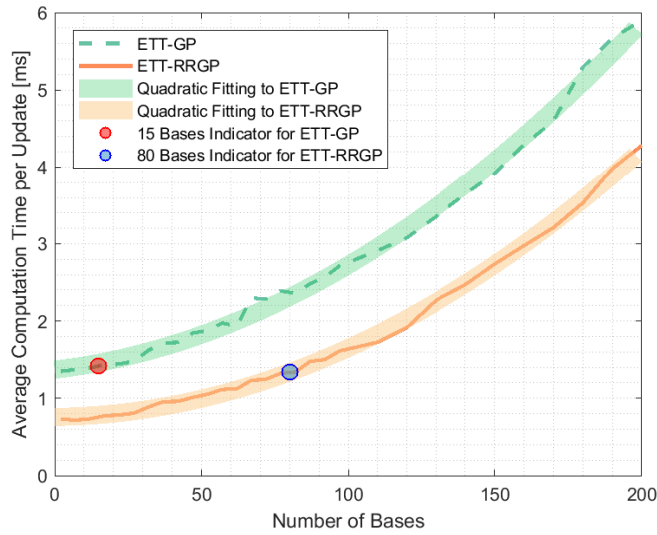
Multiplication of  $n \times m$  matrix with  $m \times p$  has  $\mathcal{O}(nmp)$  time complexity by using Schoolbook matrix multiplication algorithm, which is considered as standard matrix multiplication [61]. Moreover, the inversion of  $n \times n$  matrix with the Gauss-Jordan elimination method requires  $\mathcal{O}(n^3)$  complexity. Furthermore, addition of two  $n \times m$  matrices has  $\mathcal{O}(nm)$  time complexity since the algorithm visits each of the elements once.

Table 5.4: Time complexities in EKF measurement update equations introduced in equations 2.12a to 2.12d.  $m$  stands for the state dimension, and  $n$  is the number of measurements.

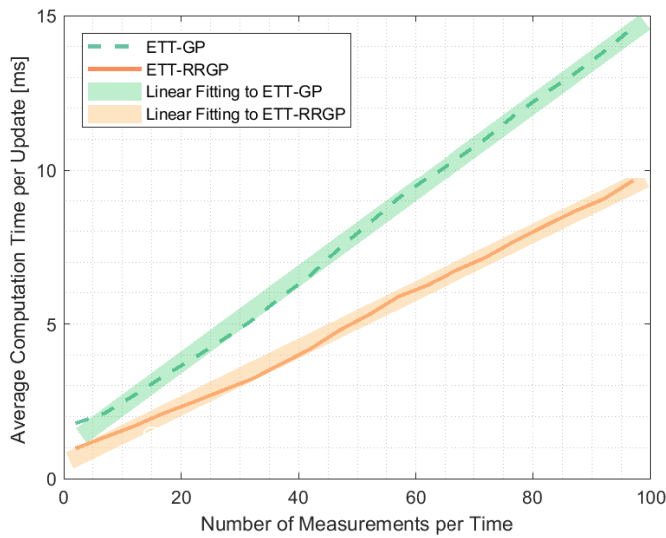
Step	Time Complexity
Innovation covariance calculation	$\mathcal{O}(\max\{nm^2, mn^2\})$
Kalman gain calculation	$\mathcal{O}(\max\{n^3, mn^2, nm^2\})$
Posterior state estimate	$\mathcal{O}(nm^2)$
Posterior covariance estimate	$\mathcal{O}(nm^2)$

Simulation computation times are highly based on the number of measurements and the number of bases as described. They are independent of the target extent or the scenario path. Simulations regarding computational aspects are conducted by obtaining average times that each method requires to complete calculations per an update.

To calculate computation times, each update time is elapsed along the trajectory, MC runs, and then averaged to present the difference. Figure 5.5 to show the effects of both the number of basis and number of measurements to computation times comparably for two approaches.



(a) Average computation time with respect to the number of bases with 10 measurements per time. Indicators demonstrate the basis number that can be used for an equal computation time.



(b) Average computation time with respect to the number of measurements with 20 number of bases.

Figure 5.5: Average computation times per update in simulations.

Figure 5.5a demonstrates average computation time per update in terms of milliseconds with regard to the number of bases. In both methods, computation times increase quadratically by increasing number of bases. The difference in the computation times between the two methods increases as well. Figure 5.5b shows the effect of the number of measurements per time on computation times. Linear growth can be seen in Figure 5.5b in both methods. The difference also has a linear behavior with regard to the number of measurements as expected. As highlighted in 5.5, ETT-RRGP works faster than ETT-GP.

### 5.3 Performance Evaluations

Assessing the performance of target tracking algorithms can be a challenging task. There is not a superior method that is capable of handling all requirements, such as kinematic error, computational complexity, or implementation difficulty. The performance may depend on the application that may be associated with specific circumstances or functional concerns. Assessment of extended target tracking algorithms can be more sophisticated since the variety of shapes expands the space.

For comparison measure, root mean square error (RMSE), also known as root mean square deviation, is regarded for both position and orientation. It is defined between the  $N$  dimensional estimated quantity  $\hat{\theta}$  and its true value  $\theta$  as

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (\theta - \hat{\theta})^2}. \quad (5.21)$$

Position RMSE is calculated firstly for two dimensional position per each trajectory time, afterward averaged for each time and then for each run. RMSE for the angle is calculated firstly for each trajectory time then averaged for MC runs. Therefore, RMSE stands for the average RMSE.

Moreover, calculating RMSE can be misleading for the target extent since it is directly related to the estimated center, orientation, and highly depends on control points. Fig-

Figure 5.6 presents this coupling with an example case where the green line corresponds to ground truth and the blue line for estimation.

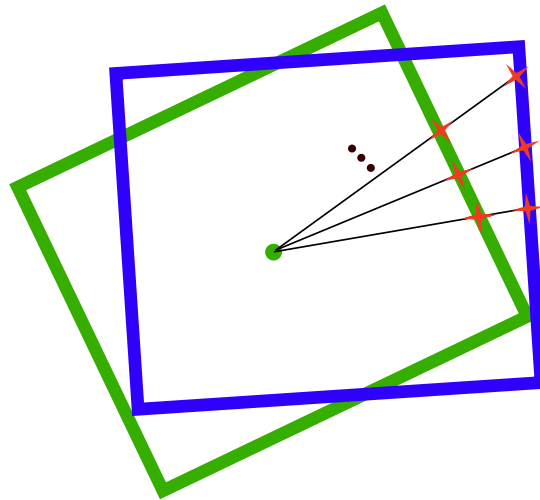


Figure 5.6: An example case with RMSE control points for the target extent

On the other hand, there is another suggestion for extended target performance evaluation metric in [24] called intersection over union (IOU), which is common in computer vision literature with an example in [2] or [52] for video tracking. IOU is also known as the Jaccard index or the Jaccard similarity coefficient in the literature [69]. It is basically the ratio between the intersection area of two objects to the union of them such that

$$\text{IOU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \in [0, 1] \quad (5.22)$$

Figure 5.7 illustrates this calculation where a green object can be considered ground truth, and the blue one is an estimation. The red wavy lines correspond to the intersection area of them, and the black diagonal lines show the union area of two objects.

Furthermore, three different possibilities for IOU are visualized in Figure 5.8 as a poor, fine, and perfect match.

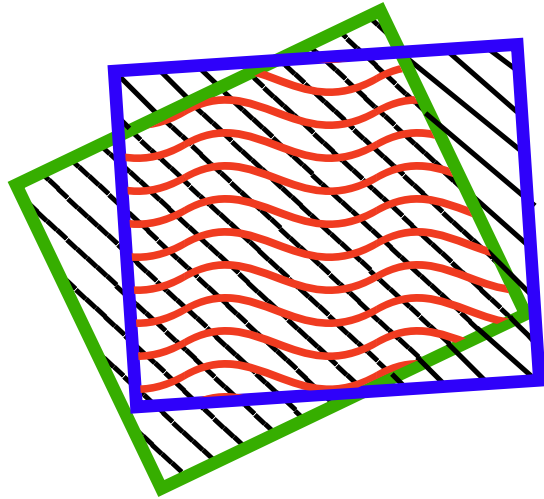
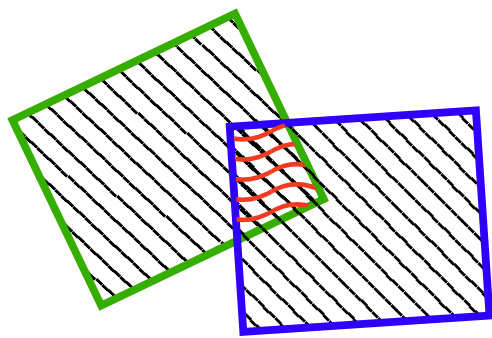
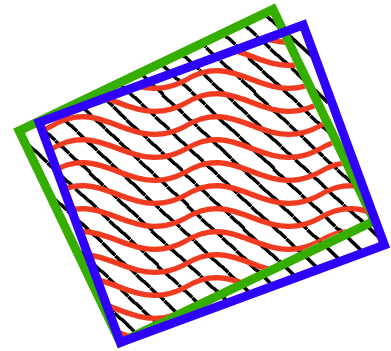


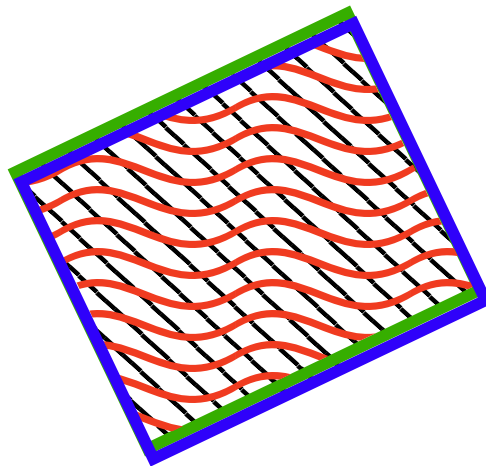
Figure 5.7: Intersection over union illustration for extended targets



(a) Poor IOU



(b) Fine IOU



(c) Perfect IOU

Figure 5.8: Example illustrations for different IOU levels

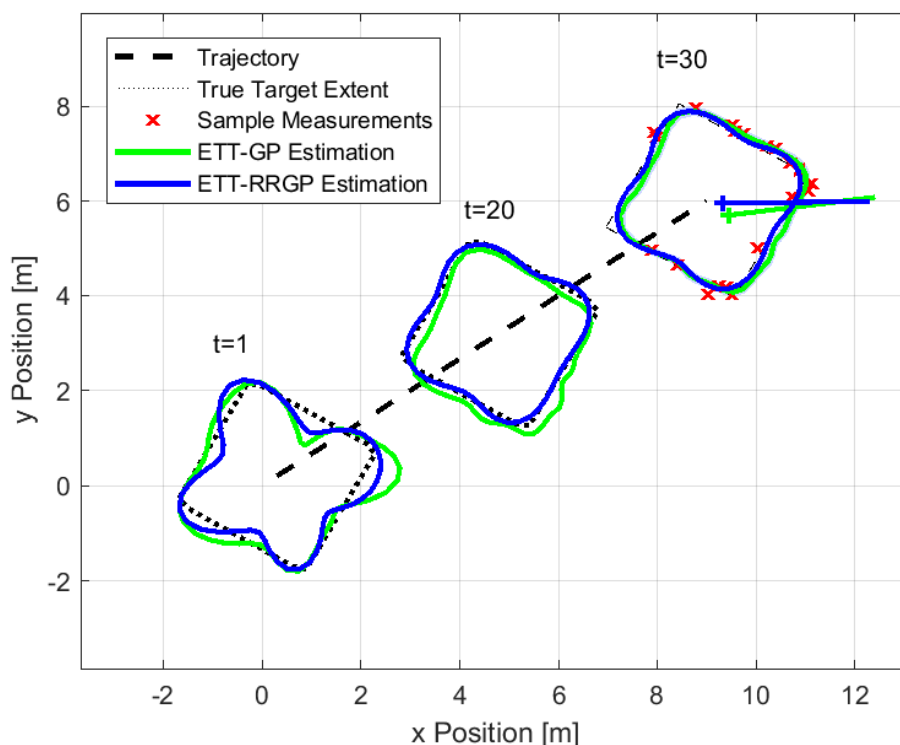


Figure 5.9: An example ETT illustration for a diagonal path with a square shape using ETT-GP and ETT-RRGP with sampled time instances

Performance comparisons can be performed for ETT-RRGP by implementing the method in [66] and obtaining results with the same measurements and the trajectory. An example of the first scenario can be seen in Figure 5.9, where the dashed diagonal line corresponds to the target center trajectory, dotted lines for the target true extent, green lines for the method in [66] called ETT-GP, and blue lines for the proposed ETT-RRGP method. In this scenario, state consists of 20 bases for both algorithms, and 20 measurements are generated by uniformly distributed locations on the target extent per trajectory time. Measurements are corrupted by Gaussian noise with the standard deviation of 0.1 meter. Figure 5.9 consists of three shape illustrations corresponding to different time instances written above for each shape. In the last time instance, measurements with red signs around the extent, orientation from the center with blue and green lines are also depicted.

The Monte Carlo simulation technique is adopted with 100 runs using random mea-

surements with the same number of measurements for each run; moreover, RMSE and IOU metrics are calculated as introduced before. Table 5.5 compares the ETT-GP method and ETT-RRGP for an example diagonal path with a square extent. ETT-RRGP is slightly better in terms of both IOU and RMSE for this scenario.

Table 5.5: RMSE and IOU to compare ETT-GP and ETT-RRGP algorithms for a square extent along a diagonal path

Method	RMSE		IOU
	Position [m]	Angle [deg]	
ETT-GP	0.21	0.49	0.938
ETT-RRGP	0.15	0.30	0.943

Simulations are also conducted for an additional scenario, which considers a triangle shape with a path that consists of a U-turn, as shown in figure 5.10. The number of measurements is chosen as 40, and 20 bases are used to track the maneuvering triangle. Noises are generated by the same parameters in all scenarios. RMSE and IOU results for a U-turn scenario are shown in table 5.6. They give similar IOU values and ETT-GP slightly better in kinematics.

Table 5.6: RMSE and IOU to compare ETT-GP and ETT-RRGP algorithms for a triangle extent along a U-turn path

Method	RMSE		IOU
	Position [m]	Angle [deg]	
ETT-GP	0.28	1.31	0.912
ETT-RRGP	0.32	4.41	0.909

Comparison results of specific scenarios may depend on several factors such as model mismatched, parameter tuning, number of bases, or measurements. Model mismatched commonly occurs when the target extent is generated from a different kernel that model considers. Other factors also expand the comparison space.

In order to express the general panorama of IOU over extents and trajectories in comparison with ETT-GP and ETT-RRGP, IOU differences of two methods are calcu-

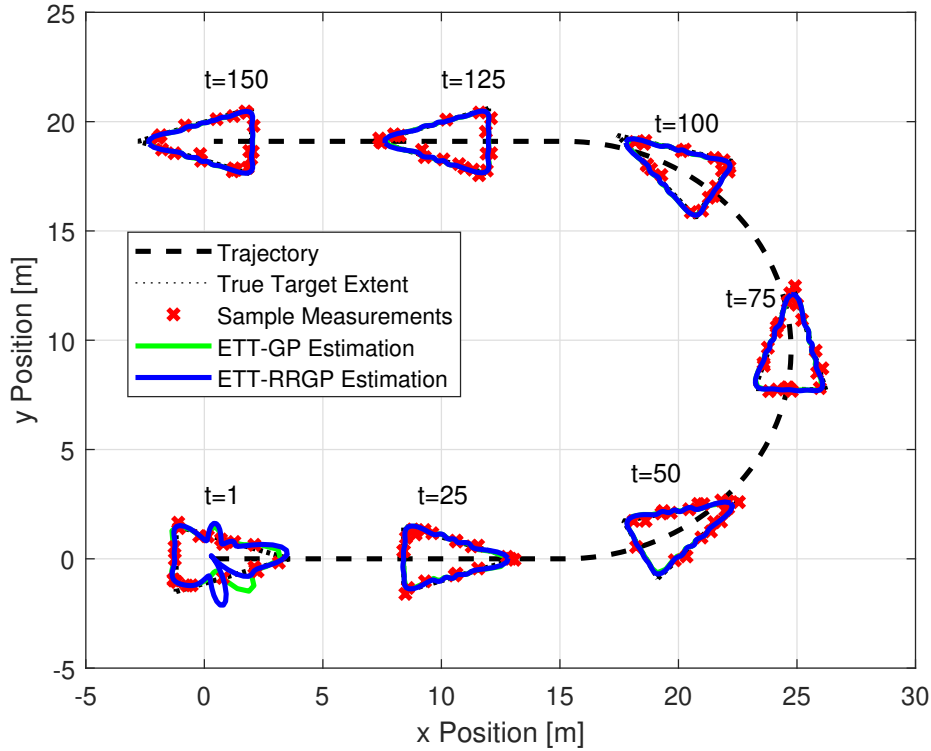


Figure 5.10: An example ETT illustration for a U-turn path with a triangle shape using ETT-GP and ETT-RRGP with sampled time instances

lated for varying number of bases and number of measurements than averaged for different shapes. Figure 5.11 shows the number of bases and measurements where ETT-RRGP's performance is better, where ETT-GP outperforms, or where there are comparable IOU results.

To investigate the areas where comparable results are confronted, other corruptive factors arise from model mismatched or tracking dynamics are tried to be minimized. To obtain IOU results without model mismatched or kinematic effects, we have taken sample functions generated by a periodic kernel that ETT methods have advanced. An example model matched extent is depicted in Figure 5.12. Simulations are conducted with known kinematics to reveal the IOU performance with respect to bases regarding two methods. Figure 5.13 shows the average IOU results of ETT-GP and ETT-RRGP with the different number of measurements per time. Full GP solution is also included for this comparison, which is introduced in section 3.

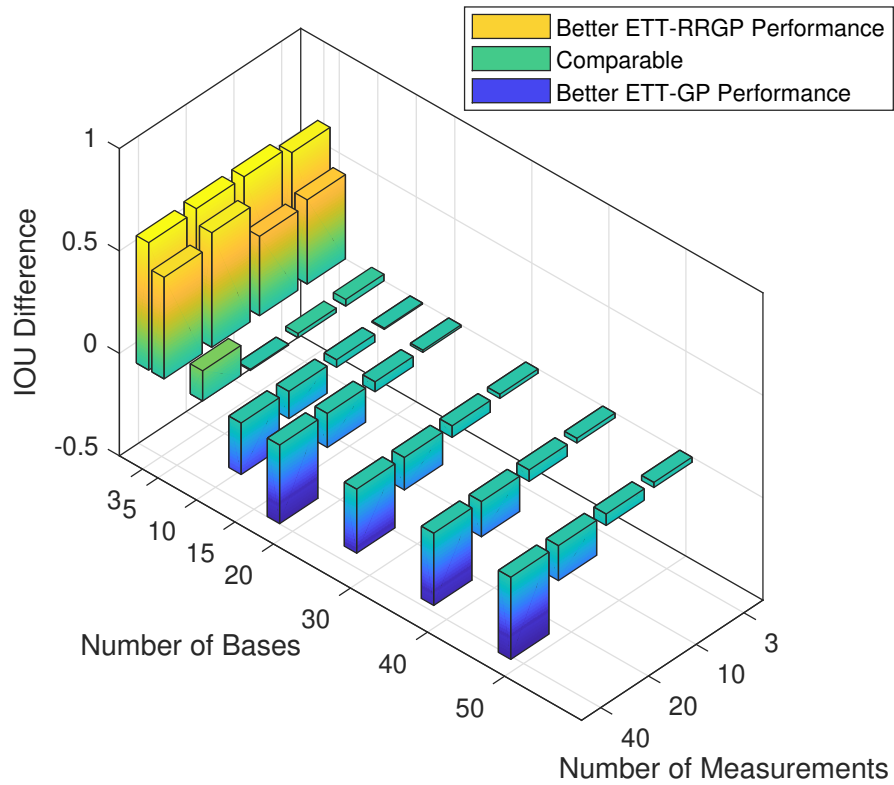


Figure 5.11: A general panorama of IOU difference between ETT-RRGP and ETT-GP with respect to the number of measurements and bases

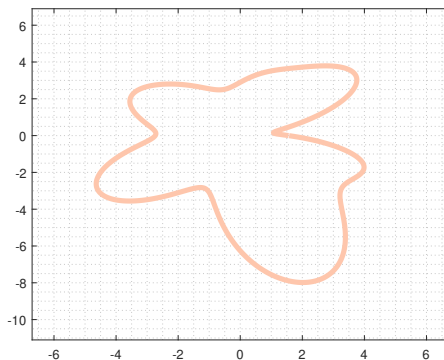
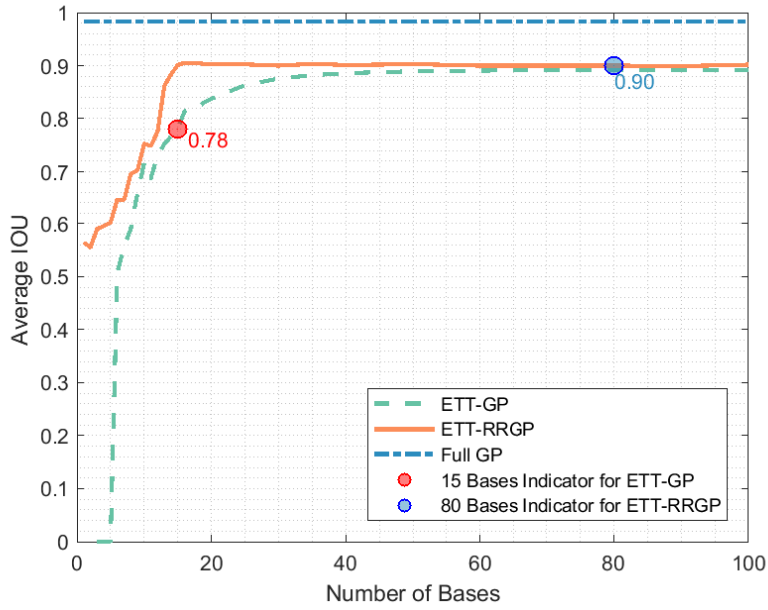
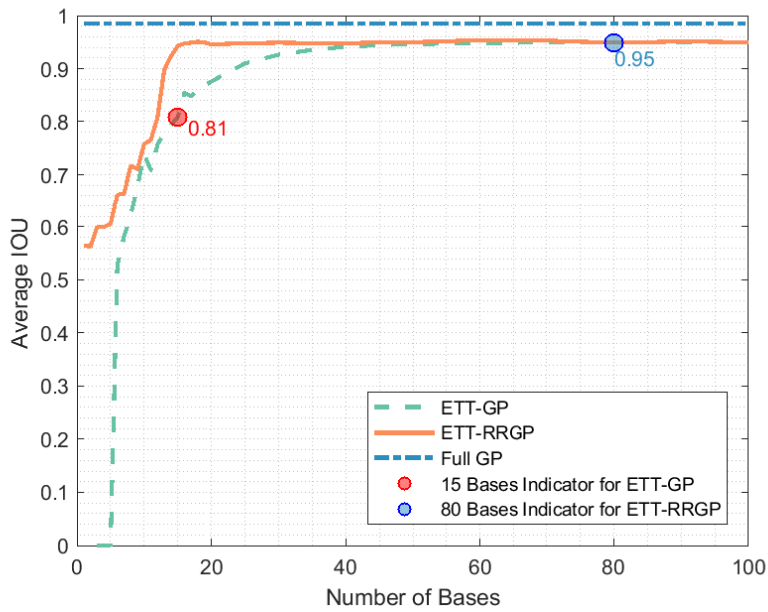


Figure 5.12: A randomly sampled extent for periodic kernel

As can be seen in figure 5.13, ETT-RRGP gives better performance for the small number of bases independent of the number of measurements. Increasing measurements per time has a positive effect on IOU results on both methods. The minimum basis point that gives the same IOU results in both methods has a negative correlation with



(a) Average IOU with 5 Measurements per time



(b) Average IOU with 10 Measurements per time

Figure 5.13: Average IOU results for model matched targets using ETT-RRGP, ETT-GP, and full GP methods with indicators for 15 and 80 bases on each comparison. Indicators emphasize that the results for the same computation time per update with different algorithms.

the number of measurements per time. Although full GP solution is optimal, it is less than the maximum IOU that can be reached. This stems from the finite number of test points.

Indicators in Figure 5.5a demonstrate the increment for the basis number that can be used for an equal computation time. ETT-RRGP can exploit roughly 80 bases in the state, whereas ETT-GP can only use 15 bases. This enhancement increases performance regarding IOU when ETT-RRGP is used, as shown in Figure 5.13a and Figure 5.13b for all measurement numbers.

A further implication can be addressed in Figure 5.13 by highlighting the minimum number of bases that is required to reach at least 98% of the maximum performance of each method. On the one hand, ETT-GP necessities using at least 30 basis points for 98% of its performance; however, on the other hand utilizing 14 bases are sufficient for the same criteria regarding ETT-RRGP.



## CHAPTER 6

### CONCLUDING REMARKS AND FUTURE WORK

On account of a target can only be seen with at most one measurement per time step from the sensor's point of view, classical target tracking algorithms render the point target assumption by ignoring its spatial extent. This has started to be out of date by recent developments in sensor technology. When different scatterers belonging to the same target give rise to distributed measurements that are generated by the target extent, researchers have addressed embedding extent information into tracking structure. Numerous suggestions have been provided in the literature about modeling the extent such as circle, rectangle, and so forth. Star-convex modeling is adopted for this thesis, and it is described with the Gaussian processes in a probabilistic framework. Utilizing GPs as a non-parametric learning method increases the amount of resources required to run the algorithm. We have tried to propose a method that is capable of handling the problem in the spectral domain apart from the approximations in the literature.

In essence, a key problem, extended target tracking, is studied throughout the paper by providing a methodology for ETT with a different framework. We have managed to make a contribution to previously suggested ETT algorithms that utilize Gaussian processes for star-convex modeling. A systematic study using the reduced rank approach, a spectral approximation method of GP regression, is proposed for ETT and alleviate computational demands in a recursive structure.

In order to represent the target extent with GP using the reduced rank approach, the necessity of attempts to express a periodic pattern is highlighted in mathematical

terms. To present a solution, a periodic adaptation of the classical reduced rank approach is carried out. This is achieved by reformulating the boundary conditions and by solving resulting differential equations explicitly. Subsequently, closed-form expressions are outlined for general restricted domains. By using these solutions, both the batch regression and the recursive regression algorithms are presented, and they are exploited in an augmented tracking structure.

In the final analyses, our method has several particular advantages. To begin with, ETT-RRGP makes fewer total number of matrix multiplications than ETT-GP. This leads to alleviating computational demands, especially for big data sets. Although it works faster, its performance is comparably successful. Secondly, the state consists of coefficients in ETT-RRGP, unlike predetermined points in ETT-GP. Therefore, predictions at desired points can be acquired just with one multiplication operation independent of the density between the prediction points. In the condition that fewer state variables have to be used, ETT-RRGP still reveals satisfactory results, and this may be considered a promising aspect of the suggested method. Lastly, a measurement is not physically associated with basis points as ETT-GP does with assumptions; it is used directly in the equations given for ETT-RRGP without any assumption.

As expected, the spectral nature of ETT-RRGP does not always lead to an advantage. One downside regarding ETT-RRGP is that increasing state dimension has limited improvements on results after a certain number of state variables compared to ETT-GP. This factor stems from its spectral distribution in which first coefficients correspond for general routine and others for details as they increase. On the other hand, when considering ETT-GP, any enhancement on state dimension increases performance as they correspond for a physical point on the target extent.

To carry our research further, we intend to make extensions of ETT-RRGP. That further research should be undertaken to investigate the effects of the reduced rank approach in higher dimensional space. Additional studies are also recommended for the reduced rank approach to examine the performance in the other areas confronted with periodic problems. On a wider level, we hope that our research will serve as a base for future works on real-time applications such as radar target classification

while tracking with its favorable computational properties.



## REFERENCES

- [1] W. Aftab, A. De Freitas, M. Arvaneh, and L. Mihaylova. A Gaussian process approach for extended object tracking with random shapes and for dealing with intractable likelihoods. In *2017 22nd International Conference on Digital Signal Processing (DSP)*, pages 1–5. IEEE, 2017.
- [2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, 2012.
- [3] D. Alspach and H. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448, 1972.
- [4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
- [5] M. Bauer, M. van der Wilk, and C. E. Rasmussen. Understanding probabilistic sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems*, pages 1533–1541, 2016.
- [6] M. Baum, F. Faion, and U. D. Hanebeck. Modeling the target extent with multiplicative noise. In *2012 15th International Conference on Information Fusion*, pages 2406–2412. IEEE, 2012.
- [7] M. Baum, M. Feldmann, D. Fränken, U. D. Hanebeck, and W. Koch. Extended object and group tracking: A comparison of random matrices and random hypersurface models. *INFORMATIK 2010. Service Science–Neue Perspektiven für die Informatik. Band 2*, 2010.

- [8] M. Baum and U. D. Hanebeck. Random hypersurface models for extended object tracking. In *2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 178–183. IEEE, 2009.
- [9] M. Baum and U. D. Hanebeck. Extended object tracking with random hypersurface models. *IEEE Transactions on Aerospace and Electronic systems*, 50(1):149–159, 2014.
- [10] M. Baum and U. D. Hanebeck. Extended object tracking with random hypersurface models. *IEEE Transactions on Aerospace and Electronic systems*, 50(1):149–159, 2014.
- [11] M. Baum, V. Klumpp, and U. D. Hanebeck. A novel Bayesian method for fitting a circle to noisy points. In *2010 13th International Conference on Information Fusion*, pages 1–6. IEEE, 2010.
- [12] Y. Bazilevs and T. J. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 36(1):12–26, 2007.
- [13] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House radar library. Artech House, 1999.
- [14] Y. Boers, H. Driessen, J. Torstensson, M. Trieb, R. Karlsson, and F. Gustafsson. Track-before-detect algorithm for tracking extended targets. *IEE Proceedings-Radar, Sonar and Navigation*, 153(4):345–351, 2006.
- [15] M. Borchers. Parametric equation of a regular pentagon. <https://www.geogebra.org/m/cXXGKUQk>. Accessed: 2020-24-12.
- [16] L. Csató and M. Opper. Sparse on-line Gaussian processes. *Neural Computation*, 14(3):641–668, 2002.
- [17] P. Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.
- [18] M. Feldmann, D. Franken, and W. Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4):1409–1420, 2010.

- [19] K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. *IEE Proceedings-Radar, Sonar and Navigation*, 152(5):364–371, 2005.
- [20] N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.
- [21] N. Graham, R. Jaffe, V. Khemani, M. Quandt, O. Schroeder, and H. Weigel. The Dirichlet Casimir problem. *Nuclear Physics B*, 677(1-2):379–404, 2004.
- [22] K. Granström and M. Baum. Extended object tracking: Introduction, overview and applications. *CoRR*, abs/1604.00970, 2016.
- [23] K. Granström and C. Lundquist. On the use of multiple measurement models for extended target tracking. In *Proceedings of the 16th International Conference on Information Fusion*, pages 1534–1541. IEEE, 2013.
- [24] K. Granström, C. Lundquist, and U. Orguner. Tracking rectangular and elliptical extended targets using laser measurements. In *14th International Conference on Information Fusion*, pages 1–8. IEEE, 2011.
- [25] K. Granström, P. Willett, and Y. Bar-Shalom. An extended target tracking model with multiple random matrices and unified kinematics. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1007–1014. IEEE, 2015.
- [26] J. Hensman, N. Durrande, and A. Solin. Variational Fourier features for Gaussian processes. *The Journal of Machine Learning Research*, 18(1):5537–5588, 2017.
- [27] Q. Hu, H. Ji, and Y. Zhang. Tracking of maneuvering non-ellipsoidal extended target with varying number of sub-objects. *Mechanical Systems and Signal Processing*, 99:262–284, 2018.
- [28] M. Isard and A. Blake. Condensation—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

- [29] A. H. Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.
- [30] C. Jidling. Tailoring Gaussian processes for tomographic reconstruction, 2019.
- [31] S. J. Julier and J. K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–193. International Society for Optics and Photonics, 1997.
- [32] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.
- [33] S. F. Kara and E. Özkan. Multi-ellipsoidal extended target tracking using sequential Monte Carlo. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2018.
- [34] H. Kaulbersch, J. Honer, and M. Baum. A cartesian B-spline vehicle model for extended object tracking. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–5. IEEE, 2018.
- [35] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.
- [36] C. Knill, A. Scheel, and K. Dietmayer. A direct scattering model for tracking vehicles with high-resolution radars. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 298–303. IEEE, 2016.
- [37] J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, 2008.
- [38] J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, 2008.

- [39] M. Kok and A. Solin. Scalable magnetic field slam in 3d using Gaussian process maps. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1353–1360. IEEE, 2018.
- [40] M. Kumru, H. Köksal, and E. Özkan. Variational measurement update for extended object tracking using Gaussian processes. Feb 2021. TechRxiv:14041778.
- [41] M. Kumru and E. Özkan. 3d extended object tracking using recursive Gaussian processes. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2018.
- [42] M. Kumru and E. Özkan. Three-dimensional extended object tracking and shape learning using Gaussian processes. 2019. arXiv:1909.11358.
- [43] J. Lan and X. R. Li. Tracking of extended object or target group using random matrix—part i: New model and approach. In *2012 15th International Conference on Information Fusion*, pages 2177–2184. IEEE, 2012.
- [44] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, 2003.
- [45] E. Maggio and A. Cavallaro. *Video tracking: theory and practice*. John Wiley & Sons, 2011.
- [46] A. J. McHutchon et al. *Nonlinear modelling and control using Gaussian processes*. PhD thesis, University of Cambridge, 2015.
- [47] R. M. Neal. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. *arXiv preprint physics/9701026*, 1997.
- [48] U. Orguner. A variational measurement update for extended target tracking with random matrices. *IEEE Transactions on Signal Processing*, 60(7):3827–3834, 2012.
- [49] E. Özkan, M. B. Guldogan, U. Orguner, and F. Gustafsson. Ground multiple target tracking with a network of acoustic sensor arrays using phd and cphd

- filters. In *14th International Conference on Information Fusion*, pages 1–8. IEEE, 2011.
- [50] E. Özkan, N. Wahlström, and S. J. Godsill. Rao-blackwellised particle filter for star-convex extended target tracking models. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1193–1199. IEEE, 2016.
- [51] A. Papoulis and S. U. Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.
- [52] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek. Video processing from electro-optical sensors for object detection and tracking in a maritime environment: a survey. *IEEE Transactions on Intelligent Transportation Systems*, 18(8):1993–2016, 2017.
- [53] J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.
- [54] C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
- [55] M. Seeger, C. Williams, and N. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. Technical report, 2003.
- [56] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2006.
- [57] A. Solin and M. Kok. Know your boundaries: Constraining Gaussian processes by variational harmonic features. *arXiv preprint arXiv:1904.05207*, 2019.
- [58] A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä. Modeling and interpolation of the ambient magnetic field by Gaussian processes. *IEEE Transactions on Robotics*, 34(4):1112–1127, 2018.
- [59] A. Solin and S. Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, pages 1–28, 2014.

- [60] J. Stewart. Second-order linear differential equations. In *Calculus*, chapter 17, pages 1193–1222. Cengage Learning, 2015.
- [61] V. Strassen. Algebraic complexity theory. In *Algorithms and Complexity*, pages 633–672. Elsevier, 1990.
- [62] L. Sun, J. Lan, and X. R. Li. Extended target tracking using star-convex model with nonlinear inequality constraints. In *Proceedings of the 31st Chinese Control Conference*, pages 3869–3874. IEEE, 2012.
- [63] L. Sun, J. Lan, and X. R. Li. Extended target tracking using star-convex model with nonlinear inequality constraints. In *Proceedings of the 31st Chinese Control Conference*, pages 3869–3874. IEEE, 2012.
- [64] S. Särkkä. *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.
- [65] B. Tuncer, M. Kumru, E. Ozkan, and A. A. Alatan. Extended object tracking and shape classification. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–5. IEEE, 2018.
- [66] N. Wahlström and E. Özkan. Extended target tracking using Gaussian processes. *IEEE Transactions on Signal Processing*, 63(16):4165–4178, 2015.
- [67] A.-M. Wazwaz. *Partial differential equations and solitary waves theory*. Springer Science & Business Media, 2010.
- [68] A. Wilson and R. Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning*, pages 1067–1075, 2013.
- [69] S. Yang, M. Baum, and K. Granström. Metrics for performance evaluation of elliptic extended object tracking methods. In *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 523–528. IEEE, 2016.
- [70] I. Yanovsky. *Partial Differential Equations: Graduate Level Problems and Solutions*. CreateSpace Independent Publishing Platform, 2014.

[71] P. Zarchan and H. Musoff. *Fundamentals of Kalman filtering: a practical approach*. American Institute of Aeronautics and Astronautics, Inc., 2013.