NEW MODELS AND INFERENCE TECHNIQUES FOR GAUSSIAN PROCESS-BASED EXTENDED OBJECT TRACKING

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

BY

MURAT KUMRU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2022

Approval of the thesis:

NEW MODELS AND INFERENCE TECHNIQUES FOR GAUSSIAN PROCESS-BASED EXTENDED OBJECT TRACKING

submitted by **MURAT KUMRU** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy 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. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** ________________

Assoc. Prof. Dr. Emre Özkan
Supervisor, **Electrical and Electronics Engineering** ________________

Examining Committee Members:

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

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

Assoc. Prof. Dr. Cem Tekin
Electrical and Electronics Engineering Department, Bilkent Uni. ________________

Assoc. Prof. Dr. Emine Ülkü Sarıtaş Çukur
Electrical and Electronics Engineering Department, Bilkent Uni. ________________

Assoc. Prof. Dr. Mustafa Mert Ankaralı
Electrical and Electronics Engineering Department, METU ________________

Date: **09.09.2022**

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, Surname: Murat Kumru

Signature :

ABSTRACT

NEW MODELS AND INFERENCE TECHNIQUES FOR GAUSSIAN PROCESS-BASED EXTENDED OBJECT TRACKING

Kumru, Murat
Ph.D., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Emre Özkan

September 2022, 164 pages

In this thesis, we consider the problem of tracking dynamic objects with unknown shapes using point cloud measurements generated by, e.g., lidars, radars, and depth cameras. The point measurements do not only convey information about the object pose, i.e., position and orientation, but they also naturally reveal the characteristics of its latent extent. Aiming to harness the full potential of the available information, we investigate the Gaussian process-based extended object tracking (GPEOT) framework.

We hereby develop several three-dimensional (3D) GPEOT models that effectively use the information provided by 3D point cloud measurements. The resulting methods can accurately estimate the 3D object shape together with its kinematic properties, such as position, orientation, and velocity. Furthermore, we introduce an approximate inference method for the GPEOT models relying on the variational Bayesian technique, where the approximate posterior distributions of the kinematic and extent variables are effectively computed by fixed-point iterations. The resulting method is particularly shown to prove robust against model uncertainties. We also focus on improving the computational characteristics of the existing GPEOT algorithms without compromising their effective performance. To this end, we formulate an alternative approximate description of the underlying GP model for the extent that provides satisfactory performance at a lower computational load. This formulation is used to derive both two- and three-dimensional tracking algorithms. Additionally, we propose a novel model that does not require the star-convexity assumption, as opposed to the standard GPEOT. Therefore, this formulation expands the application of the existing GPEOT framework as it enables tracking arbitrarily-shaped objects while learning their latent extent. Comprehensive experiments are performed to demonstrate the added value of the mentioned contributions with both simulated and real measurements.

Keywords: Extended Object Tracking, Object Tracking, Gaussian Processes, Variational Bayes.

ÖZ

GAUSS SÜREÇ TABANLI GENİŞLETİLMİŞ CİSİM TAKİBİ İÇİN YENİ MODELLER VE KESTİRİM YÖNTEMLERİ

Kumru, Murat
Doktora, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Emre Özkan

Eylül 2022 , 164 sayfa

Bu tezde, bilinmeyen şekilli dinamik nesnelerin, -lidar, radar ve derinlik kamerası gibi algılayıcılar tarafından üretilen- nokta bulutu ölçümleri kullanılarak takip edilmesi problemi ele alınmıştır. Nokta ölçümleri, nesnelerin konumları ve yönelimlerinin yanında, bilinmeyen şekilleri hakkında da önemli bilgiler taşırlar. Bu bilgilerin potansiyelinden tam olarak faydalanabilmek amacıyla, Gauss süreç tabanlı genişletilmiş nesne takibi (GPEOT) çerçevesi incelenmiştir.

Bu bağlamda, üç boyutlu (3B) nokta bulutu ölçümleri tarafından sağlanan bilgileri etkin bir biçimde kullanabilen çeşitli 3B GPEOT modelleri önerilmiştir. Ortaya çıkan yöntemler, cismin konum, yönelim ve hız gibi kinematik özellikleri ile birlikte 3B uzantısını da yüksek doğrulukla kestirebilmektedir. Öte yandan, GPEOT modelleri için varyasyonel Bayes tekniğine dayanan yaklaşık bir kestirim metodu türetilmiştir. İlgili yöntem, cismin kinematik ve uzantı değişkenlerininin yaklaşık sonsal dağılımlarını sabit nokta yinelemeleri ile başarılı bir şekilde hesaplamaktadır. Geliştirilen takip algoritmasının özellikle model belirsizliklerine karşı gürbüz olduğu gösterilmiştir. Ayrıca, mevcut GPEOT algoritmalarının etkin performanslarından ödün vermeden hesaplama özelliklerini iyileştirmeye odaklanılmıştır. Bu amaçla, cismin uzantısını açıklamak üzere kullanılan Gauss süreci modeli alternatif bir yöntem ile yaklaştırılarak gereken işlem yükü azaltılmıştır. Bu formülasyon kullanılarak iki ve üç boyutlu hedef takibi algoritmaları türetilmiştir. Ek olarak, standart GPEOT yaklaşımının aksine, yıldız-dışbükey varsayımına dayanmayan yeni bir uzantı modeli geliştirilmiştir. Böylece, rastgele uzantılı cisimlerin GPEOT çerçevesinde takibi ve eş zamanlı olarak uzantılarının öğrenilmesi mümkün kılınmıştır. Bahsedilen çalışmaların katma değeri, benzeştirilmiş ve gerçek ölçümler üzerinde gerçekleştirilen kapsamlı deneyler ile gösterilmiştir.

Anahtar Kelimeler: Genişletilmiş Cisim Takibi, Cisim Takibi, Gauss Süreçleri, Varyasyonel Bayes.

To my dear family and my lovely wife Elif

ACKNOWLEDGMENTS

I believe every PhD journey is challenging in its own way, and this specific work was no exception in this regard. But I was exceptionally lucky to have Emre Özkan as my supervisor, who has made this ride much smoother through his continuous supervision and encouragement. I would like to genuinely thank him for offering me the intellectual freedom to pursue my own research path while meticulously guiding me along the way.

I would like to extend my sincere thanks to Umut Orguner and Cem Tekin, who have closely followed the progress of this thesis work and contributed with their constructive criticism and invaluable insight into the problems I have tackled. I would also like to thank Emine Ülkü Sarıtaş Çukur and Mustafa Mert Ankaralı for taking part in my thesis exam committee and for their positive feedback and helpful suggestions.

During my years of assistantship in the department, I have had the chance to meet many respected faculty members who have deeply impacted and inspired me. I am truly thankful to all of them; in particular, I would like to express my gratitude to Afşar Saranlı, Arzu Tuncay Koç, Mustafa Mert Ankaralı, Mübeccel Demirekler, Ozan Keysan, and Sezai Emre Tuna for their assistance and friendly attitude.

It was a great pleasure to be a member of Sensor Fusion Laboratory and work together with great teammates. I am sincerely grateful to Barkın Tuncer, Mustafa Buğra Özcan, and Hilal Köksal for our fruitful collaborations and their warm friendship.

During the last year of my PhD, I have had the privilege of working in an amazing environment at Kuartis, surrounded by nothing but kindness and understanding. I would like to recognize all of my dear colleagues; I must specifically mention Berker Loğoğlu, Alper Küçükkömürler, Eren Şener, and Cahit Yıldırım for their constant support throughout the writing phase of this thesis.

While trying to find my way through all the ups and downs of the PhD voyage, the

comfort and relief offered by my dear friends have been invaluable. I would like to thank and acknowledge Cumhur Çakmak, İzzet Kağan Erünsal, Kenan Ahıska, Selim Özgen, Ayşe Deniz Duyul, Beril Beşbınar, Ayşegül Kılıç, Osman Kaan Tokgöz, and Onur Cem Erdoğan.

I want to express my genuine appreciation to my family. I am indebted to Selen Naz, Selime and Osman Sarıtaş, and Ezgi and Fırat Kumru for all their compassion and generosity. And, of course, I owe my deepest gratitude to my loving parents, Fürüzan and Erdal Kumru. I cherish their unconditional love and care.

Finally, I have had the immense fortune to experience every phase of this bumpy ride along with my fantastic wife, Elif. I heartily know that it would not have been possible without her support and delicate nurturing. I am truly blessed for all the joy, love, and peace she brings to my life. I am eagerly looking forward to exploring together what the future holds for us.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	Two-Dimensional
3D	Three-Dimensional
EKF	Extended Kalman Filter
EOT	Extended Object Tracking
GP	Gaussian Process
GPEOT	Gaussian Process-Based Extended Object Tracking
ICP	Iterative Closest Point
IOU	Intersection over Union
IEKF	Iterated Extended Kalman Filter
Lidar	Light Detection and Ranging
Radar	Radio Detection and Ranging
RMSE	Root-Mean-Square Error

CHAPTER 1

INTRODUCTION

1.1 Motivation

With the ever-increasing use of technology, machines are expected to perform more sophisticated means of interaction with their environment. Among the challenges of developing such systems, one is particularly difficult to overcome: designing perception systems. A perception system generally refers to a collection of sensors and algorithms processing the provided data to obtain representations of the environment and itself. Considering any decision-making process depends directly on this representation, the performance of the perception system naturally sets limits on the tasks that a machine can achieve. In this respect, with the proliferation of low-cost, high-precision sensors, and cheap computational power, there is an urgent need for improvement in the efficiency and efficacy of the existing algorithms.

Drawing on these observations, in this thesis, we aspire to contribute to a specific component of the perception systems. In particular, we develop algorithms to extract refined knowledge about dynamic objects by processing point cloud measurements, which are generated by, for example, lidars, radars, and depth cameras. More specifically, we propose methods that can track the dynamic behavior of the objects and learn their latent shape simultaneously.

1.2 Problem Definition

Object tracking is the problem of estimating the unknown motion variables, e.g., position, orientation, and their time derivatives, of dynamic objects using noisy observations acquired by, for example, cameras, lidars, and radars. It is one of the key factors that sets the limit for the overall system performance in many applications, such as autonomous driving, mobile robotics, surveillance, and aviation.

Object tracking algorithms can be grouped into two main categories. The algorithms in the first group are based on the assumption that an object can generate at most one measurement per sensor scan. Accordingly, they model the objects of interest as point sources [1, 2]. This approach is referred to as *point object tracking* in the literature. With the increasing resolution of sensors and changing characteristics of practical applications, where the sensors are located closer to their interests, a target can potentially originate multiple measurements at a single scan. The dedicated solutions addressing this problem constitute the second group, which is called *extended object tracking* (EOT) [3]. These methods aim at estimating the target's extent along with its kinematics.

A solid body of EOT literature relying on various extent representations has been developed, [3]. These representations exhibit significant variance in their compactness and expressive power. For example, a group of EOT algorithms imposes simple shape models such as a circle, a rectangle, or an ellipse. These essentially achieve extent modeling with only a few parameters at the cost of limited potential for shape description. A substantial fraction of the studies addressing problems in mobile robotics and autonomous driving resides in this category as they utilize the bounding box model. [4] considers pedestrian tracking by processing a partition of three-dimensional (3D) point cloud data. They model the extent of a pedestrian by a bounding box. In [5], tracking for autonomous driving in urban settings is considered. They formulate the problem in two-dimensional (2D) motion space regarding the bounding box model for objects. [6] offers a framework for object detection and tracking in 3D range data. While they provide a flexible object detection approach to generalize to a wide range of shapes, they prefer bounding box representation in the extended tracker. With the aim of detection and tracking of dynamic objects, [7] makes use of data from multiple sensors, e.g., radar, camera, and lidar. They preprocess 3D point cloud data to extract features, such as line segments and L-shapes, which are in turn, provided to the estimation algorithm. Objects are represented by 3D bounding boxes, while the motion is confined to 2D in this model.

Another popular line of research, named random matrix approach, essentially approximates the object extent by an ellipse, [8], [9], [10], [11]. Random hyper-surface models (RHM), on the other hand, formulates the EOT problem via a more flexible extent representation for star-convex objects, [12], [13]. The model is based on the Fourier series expansion of the spatial extent, and the coefficients of the expansion are estimated together with the kinematics. A specific adaptation of the RHM idea to tackle people tracking using depth data is presented in [14], and therein 3D shape is approximated as a cylinder. From a similar perspective, a more general tracking framework based on the assumption that a 3D object surface can be constructed by some transformations, e.g., translation, rotation, of a plane curve is proposed in [15]. However, this approach necessitates a special formulation of the recursive estimator in accordance with the particular transformation considered. This devalues the virtue of the model for a standard tracking application as there is typically no prior information about the object's shape.

An alternative approach for tracking is to make use of a grid representation, which approximates the continuous space as a collection of small-sized units. For instance, so-called occupancy grids, which were initially motivated by the mapping problem, have been adapted to tracking dynamic objects. In [16] and [17], local occupancy grids, which are fixed to the local coordinate frame of the object, are utilized for tracking arbitrarily shaped objects. Although this representation can potentially lead to a rich description of the latent shape, it is limited by the inherent assumption that the individual cells are mutually independent. In particular, this assumption possibly causes the inference to disregard the consistent spatial patterns which are locally intrinsic in the underlying object shape. Similarly, in processing environmental representations expressed by grids, neural networks are also applied to object tracking in [18], [19], [20]. All of these methods come with a fundamental trade-off between spatial resolution and memory consumption/computational load. Consequently, a vast majority of the existing literature alleviates these issues by formulating the tracking problem in 2D space. Additionally, another significant inconvenience associated with the grid-based approach is the selection of grid size and cell resolution without having a priori information about the object.

All of the approaches discussed so far prescribe object tracking to infer some latent variables which are meant to express the object extent, e.g., Fourier coefficients, random matrices, and occupancy values. In contrast, it is also possible to refer to an intuitive extent description which is formed as a collection of measurements acquired successively. Using this notion, [21] suggests to jointly estimate the self-motion and the track's motion using the Iterative Closest Point (ICP) algorithm and a Kalman filter where the appearance of the object is stored an aggregation of lidar measurements and corresponding features. [22] basically combines lidar data and color information and offers an inference method named annealed dynamics histograms based on the iterative sampling of the state-space. They also do not model the shape explicitly but instead integrate measurements over time to obtain a point cloud representation of the shape. Likewise, the object extent is expressed as an accumulated point cloud in [23]. As a principal difference, they formulate tracking as a batch optimization on a sliding window of measurements rather than applying a standard Bayesian estimator. In [24], lidar measurements are processed to jointly estimate the motion state and the shape of the object, which is expressed by means of a set of so-called boundary points. At each iteration, a laser measurement returned from the object is either interpreted as a newly observed boundary point and appended to the shape representation or associated with one of the existing boundary points and used to update it. A simple variant of the ICP algorithm is employed to find the correspondence between the stored boundary points and fresh measurements. These methods facilitate joint tracking and shape learning of arbitrary objects in the presence of continuously available, high-precision, and informative measurements. However, as they do not feature a principled representation of the underlying shape, there arise robustness issues with the sparsity of the measurements due to increasing distance, change of the vantage point and occlusions. In addition, dependence on the ICP algorithm to align point clouds renders the tracking performance sensitive to initialization errors. Lastly, the storage and computational requirements scale with the size of the object extent.

With its favorable analytical properties and close connections to the Bayesian paradigm, a Gaussian process (GP) facilitates the modeling of unknown functions. With this in mind, the authors describe latent extents of star-convex objects by GP in [25] and [26]. These models estimate the pose of the object while learning its arbitrary shape simultaneously. This approach is applied to the multi-object tracking problem in [27]. Several adaptations of the GP model are also investigated for specific application settings. Multiple sensor fusion problem in automotive scenarios is addressed in [28]. [29] suggests a tracking filter processing measurements from a high-resolution automotive radar; it is built upon the GP extent representation. [30] focuses on object classification, making use of the extent estimates produced by a GP-based tracking algorithm. In [31], measurement models to leverage both negative and positive information, i.e., where the object should and should not exist, are developed with the aim of effectively employing the laser range scanners in tracking. These models essentially exploit the GP-based description of the object shape.

1.3 Contributions and Publications

In this thesis, we focus our attention on developing effective and efficient Gaussian process-based extended object tracking (GPEOT) algorithms. The contributions provided by each chapter of the thesis are listed in the following.

- Chapter 3 investigates the problem of tracking objects with unknown shapes, particularly using 3D point cloud data. We propose a GPEOT model to jointly estimate object kinematics, including position, orientation, and velocities, together with the shape of the object for online and offline applications. We describe the unknown shape by a radial function in 3D and induce a correlation structure via a GP. Furthermore, we propose an efficient algorithm to reduce the computational complexity of working with 3D data. This is accomplished by casting the tracking problem into projection planes that are attached to the object's local frame. The resulting algorithms can process 3D point cloud data and accomplish tracking of a dynamic object. Moreover, they provide analytical expressions for the representation of the object shape in 3D, together with confidence intervals. The confidence intervals, which quantify the uncertainty in the shape estimate, can later be used for solving the gating and association problems inherent in object tracking. The performance of the methods is demonstrated both on simulated and real data. The results are compared with an existing random matrix model, which is commonly used for extended object tracking in the literature.

- Aspiring to improve the estimation performance of the GPEOT models, we suggest an alternative inference framework for the GPEOT models in Chapter 4. The method provides an approximate solution to the Bayesian filtering problem in GPEOT by relying on a new measurement update, which we derive using variational Bayes techniques. The resulting algorithm effectively computes approximate posterior densities of the kinematic and the extent states. We conduct various experiments on simulated and real data and examine the performance compared with a reference method, which employs an extended Kalman filter for inference. The proposed algorithm is shown to significantly improve the accuracy of both the kinematic and the extent estimates and proves robust against model uncertainties.

- In Chapter 5, we propose a new formulation to the GPEOT algorithms that leads to improved computational characteristics. The GPEOT models describe the target extent by a radial distance function, which is probabilistically modeled by a GP. To attain practically feasible trackers, which recursively process collected measurements, the existing GPEOT methods utilize an inducing point approximation for the original GP model. In this scheme, the extent is expressed by a set of radial function values, which are typically evaluated at a uniformly spaced grid. We hereby employ an alternative spectral-domain approximation for the underlying GP model. This method offers an approximate basis function expansion, where the weights are random variables with Gaussian distribution. The weight vector essentially forms a parametrized description of the target extent. We construct state-space models that include the kinematics and the weight vector for both 2D and 3D problems. Then, it is possible to estimate the state vector recursively by standard Bayesian techniques that constitutes an EOT algorithm. We illustrate an efficient implementation by relying on the extended Kalman filtering. The key contribution of our approach is the reduction in the computational complexity of the existing GPEOT models without compromising effective performance. We analytically investigate the computational properties of the proposed algorithms. Additionally, the performance of the suggested trackers is examined with comprehensive simulation experiments.

- Aiming to generalize the GPEOT framework to a broader class of objects, we suggest a novel model in Chapter 6. The derivation of the existing GPEOT algorithms is based on the assumption that the object of interest is star-convex. In this manner, it becomes possible to describe the latent extent by a radial distance function, which is modeled by a GP. To further improve the flexibility of the resulting trackers, we propose to employ a potential function to indicate the unknown object extent. This formulation is capable of representing arbitrarily-shaped objects, which might be non-convex and consist of disconnected subparts. Closely following the original idea of GPEOT, the potential function is then modeled by a GP, which systematically accounts for the intrinsic spatial correlation of the extent. Subsequently, we construct a state-space model regarding both the kinematic variables and an approximate description of the underlying GP model. One can estimate the state vector via standard Bayesian techniques, which establishes an EOT algorithm. We demonstrate that the derived tracker is able to capture detailed estimates of the extent of arbitrarily-shaped objects through simulation experiments.

The list of publications that have been produced as a product of this thesis work is given below.

- M. Kumru and E. Özkan, “3D Extended Object Tracking Using Recursive Gaussian Processes," Proceedings of the International Conference on Information Fusion, 2018, [32].

- M. Kumru and E. Özkan, “Comments on ‘Spatio-Temporal Gaussian Process Models for Extended and Group Object Tracking with Irregular Shapes’," arXiv preprint arXiv:2002.08065, Feb. 2020, [33].

- M. Kumru and E. Özkan, “Three-Dimensional Extended Object Tracking and Shape Learning Using Gaussian Processes," IEEE Transactions on Aerospace and Electronic Systems, vol. 57, no. 5, pp. 2795-2814, Mar. 2021, [34].

- M. Kumru, H. Köksal and E. Özkan, “Variational Measurement Update for Extended Object Tracking Using Gaussian Processes," IEEE Signal Processing Letters, vol. 28, pp. 538-542, Feb. 2021, [35].

Moreover, the manuscripts that are in preparation for submission are given in the following.

- M. Kumru and E. Özkan, “Tracking Arbitrarily-Shaped Extended Targets Using Gaussian Processes," to be submitted to IEEE Robotics and Automation Letters.
- Buğra Özcan, M. Kumru and E. Özkan, “Extended Object Tracking Using Reduced-Rank Gaussian Processes," to be submitted to IEEE Transactions on Robotics.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Bayesian Inference

Bayesian inference is essentially the mathematical mechanism addressing the estimation of unknown parameters/variables within a Bayesian framework. It relies on modeling the uncertainties by probability distributions and using the calculus of probability. More specifically, the information provided by the observed variables is incorporated into our prior belief by means of the Bayes' rule

$$p(x|y) = \frac{p(y|x)\, p(x)}{p(y)}, \tag{2.1}$$

where x is the unknown parameter, and y is the observation. $p(x)$ is called the *prior distribution*, and it describes the preliminary information about the possibility of the parameter values. $p(y|x)$ is the *likelihood function*; it describes the probabilistic relation between the parameter and the observation. $p(y)$ is called the *normalizing constant* or the *evidence*, and it is not function of x as also revealed by the following expression $p(y) = \int p(y|x)p(x)\mathrm{d}x$. $p(x|y)$ is named as the *posterior distribution*; it represents out refined knowledge about the parameter x in accordance with the observation y. Some of the reference books that present an in-depth discussion on the Bayesian inference are [1, 36, 37, 38].

2.1.1 Sequential Bayesian Inference

In this thesis, we primarily focus on the estimation of the unknown states of various dynamic systems by processing the measurements acquired by a sensor device at discrete time instants. Such a system can be described by a discrete-time probabilistic state-space model in its most general form as in the following,

$$\mathbf{x}_{k+1} = \mathbf{f}\left(\mathbf{x}_k, \boldsymbol{\omega}_k\right), \qquad \boldsymbol{\omega}_k \sim p(\boldsymbol{\omega}_k), \tag{2.2}$$

$$\mathbf{y}_k = \mathbf{h}\left(\mathbf{x}_k, \boldsymbol{\nu}_k\right), \qquad \boldsymbol{\nu}_k \sim p(\boldsymbol{\nu}_k), \tag{2.3}$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the state vector; the state transition function $\mathbf{f}(\cdot)$ defines the evolution of the state in time, and $\boldsymbol{\omega}_k$ is the process noise that accounts for the imperfections in our process model. $\mathbf{y}_k \in \mathbb{R}^{n_y}$ indicates the measurements acquired at time k; $\mathbf{h}(\cdot)$ is the measurement model that relates the state vector to the observations, and $\boldsymbol{\nu}_k$ is the measurement noise denoting the nonideal characteristics of the sensor device. Moreover, it is typical to assume that the process and the measurement noises are white, and they are independent from the prior distribution $\mathbf{x}_0 \sim p(\mathbf{x}_0)$. Under this assumption, the resulting state-space model is Markovian such that the states constitutes a Markov sequence, and the measurement $\mathbf{y}_k$ conditioned on $\mathbf{x}_k$ is independent of the other measurements and the states, i.e., $\mathbf{x}_k$, $p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \mathbf{x}_{1:k}) = p(\mathbf{y}_k|\mathbf{x}_k)$.

Regarding the given state-space model, the objective is to compute the conditional distribution of the state vector at time k, $p(\mathbf{x}_k|\mathbf{y}_{1:N})$ where $\mathbf{y}_{1:N} \triangleq \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\}$. Depending on the time range of the available measurements, this problem is referred to as

$$\begin{cases} N < k : \text{Prediction}, \\ N = k : \text{Filtering}, \\ N > k : \text{Smoothing}. \end{cases} \tag{2.4}$$

For an online application where measurements are gathered sequentially, the batch application of the Bayesian inference is practically infeasible as the number of observations grows over time without a limit, and hence it becomes computationally intractable at some point. In this regard, the underlying idea of the sequential Bayesian inference is to treat the posterior of the previous time step as the prior of the current time step. In this scheme, the posterior is obtained in a recursive manner by only processing newly acquired observations, and the computational complexity of the resulting recursion is kept fixed for each step over time.

2.1.1.1 Bayesian Filtering

The aim in Bayesian filtering is to compute the posterior distribution of the state vector at time k by using all information provided by the measurements collected up to time k, $p(\mathbf{x}_k|\mathbf{y}_{1:k})$. For a Markovian model, the basic probability theory offers a recursive solution consisting of two successive phases, namely the prediction and the measurement update.

$$\boxed{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})} \xrightarrow{\text{prediction}} \boxed{p(\mathbf{x}_k|\mathbf{y}_{1:k-1})} \xrightarrow{\text{update}} \boxed{p(\mathbf{x}_k|\mathbf{y}_{1:k})} \tag{2.5}$$

Prediction/ Time Update:

In this step, the predicted distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ is computed from the previous posterior distribution $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ regarding the process model.

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})\mathrm{d}\mathbf{x}_{k-1} \tag{2.6}$$

$$= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \cancel{\mathbf{y}_{1:k-1}})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})\mathrm{d}\mathbf{x}_{k-1} \tag{2.7}$$

$$= \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})\mathrm{d}\mathbf{x}_{k-1} \tag{2.8}$$

In the literature, this is referred to as the Chapman-Kolmogorov equation.

Measurement Update:

After attaining the predicted distribution, the next step is to update our knowledge about $\mathbf{x}_k$ by incorporating the new measurement $\mathbf{y}_k$, which amounts to computing the estimated distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ as

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{y}_{1:k}) &= \frac{p(\mathbf{y}_k|\mathbf{x}_k, \cancel{\mathbf{y}_{1:k-1}})p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \\ &\propto p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1}), \end{aligned} \tag{2.9}$$

where $p(\mathbf{y}_k|\mathbf{x}_k)$ and $p(\mathbf{y}_k|\mathbf{y}_{1:k-1})$ are called the measurement likelihood and predicted measurement distribution, respectively.

Consequently, the algorithm to realize Bayesian filtering is summarized below.

- Start with $p(\mathbf{x}_0)$
- For each k

– Perform the prediction/ time update by

$$p\left(\mathbf{x}_k|\mathbf{y}_{1:k-1}\right) = \int p\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right) p\left(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}\right) \mathrm{d}\mathbf{x}_{k-1}$$

– Perform the measurement update by

$$\frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}$$

These formulae are valid for all probabilistic state-space models with Markovian property. In Fig. 2.1, the predicted and posterior distributions[1] are illustrated at four successive time instants for the following system, with nonlinear system dynamics and additive Gaussian noise in the process and the measurement model,

$$\begin{aligned} f(x_k) &= \cos\left(\frac{2\pi}{4}x_k\right) + \sin\left(\frac{2\pi}{6}x_k\right) + \sin\left(\frac{2\pi}{6}x_k\right)^2 + \omega_k,\ \omega_k \sim \mathcal{N}(0, 0.2), \\ h(x_k) &= x_k + \nu_k,\ \nu_k \sim \mathcal{N}(0, 3), \qquad (2.10) \\ x_0 &\sim \mathcal{N}(5, 1). \end{aligned}$$

Unfortunately, there are quite a few special cases where the posterior distribution is analytically tractable. Probably the most popular of these is that the state-space model is linear and Gaussian, which is detailed in the next section.

2.1.2 Linear Gaussian Model

The linear Gaussian model is specified by the following set of equations,

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + \boldsymbol{\omega}_k, \qquad \boldsymbol{\omega}_k \sim \mathcal{N}(\boldsymbol{\omega}_k; 0, Q), \qquad (2.11)$$

$$\mathbf{y}_k = C\mathbf{x}_k + \boldsymbol{\nu}_k, \qquad \boldsymbol{\nu}_k \sim \mathcal{N}(\boldsymbol{\nu}_k; 0, R), \qquad (2.12)$$

$$\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}_0, P_0). \qquad (2.13)$$

In this system, the state transition function and the measurement function are linear mappings of the state vector; the process and the measurement noises are additive and Gaussian distributed. Furthermore, the prior distribution is modeled to be Gaussian.

[1] In fact, the predicted and the posterior distributions are approximately computed by a point mass filter, [39].

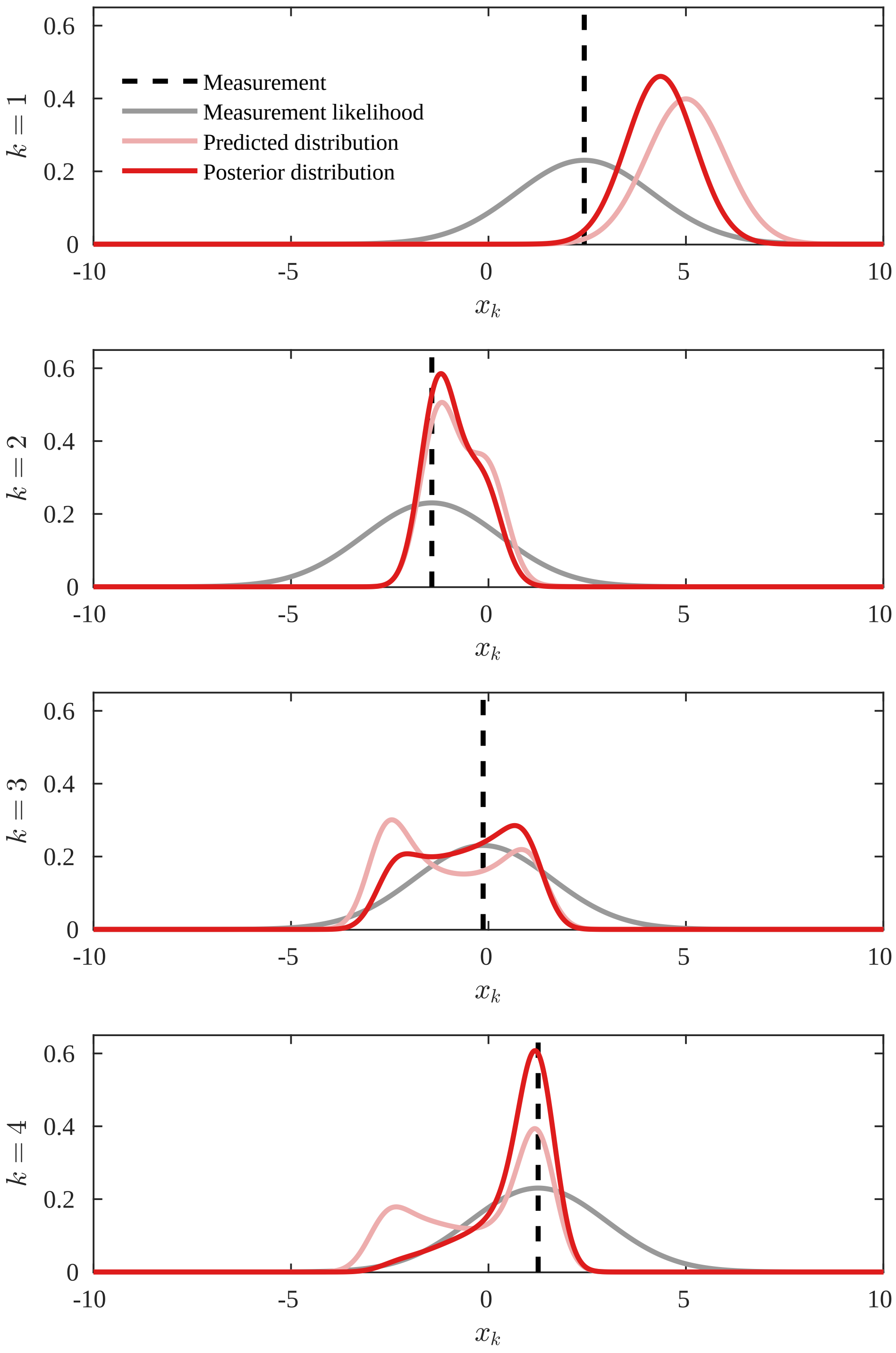


Figure 2.1: An illustration of the distributions obtained during four successive time instants of Bayesian filtering for the nonlinear system in (2.10).

In this special case, by the virtue of the favorable analytical properties of the Gaussian density, the following distributions compute as

$$\begin{aligned} p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) &= \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}), \\ p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}) &= \mathcal{N}(\mathbf{y}_k; \hat{\mathbf{y}}_{k|k-1}, S_{k|k-1}), \\ p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) &= \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, P_{k|k}). \end{aligned} \tag{2.14}$$

As an example, in Fig. 2.2, we exhibit the predicted and posterior distributions for four consecutive time steps regarding the following linear Gaussian system,

$$\begin{aligned} f(x_k) &= -0.5x_k + \omega_k,\ \omega_k \sim \mathcal{N}(0,1), \\ h(x_k) &= x_k + \nu_k,\ \nu_k \sim \mathcal{N}(0,2), \\ x_0 &\sim \mathcal{N}(5,4). \end{aligned} \tag{2.15}$$

Consequently, it is sufficient to keep and manipulate the means and covariances of the distributions during a Bayesian recursion. In particular, we can achieve this in the following manner.

$$\boxed{\hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1}} \xrightarrow{\text{prediction}} \boxed{\hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}} \xrightarrow{\text{update}} \boxed{\hat{\mathbf{x}}_{k|k}, P_{k|k}} \tag{2.16}$$

The algorithm to realize this mechanism is known as the Kalman filter, [40].

2.2 Kalman Filter

The psuedocode of the Kalman filter is given in Algorithm 1. The variables used in the equations are explicitly indicated in Table 2.1.

2.3 Variational Bayesian Inference

As previously mentioned, it is analytically infeasible to compute the exact posterior distribution of the latent variables for many probabilistic models. There is a plethora of methods in the literature that addresses this problem from different perspectives. In this section, we briefly introduce the variational Bayesian inference that seeks for approximate solution by relying on the optimization of an informative functional of the true posterior. More specifically, the objective is to approximate the true posterior distribution $p(\mathbf{x}|\mathbf{y})$ by a distribution $q(\mathbf{x})$, where $\mathbf{x}$ and $\mathbf{y}$ represent the latent variables

Algorithm 1 Pseudocode of the Kalman Filter

```
Initialize the density of the state $p(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \hat{\mathbf{x}}_{0|0}, P_{0|0})$
for time $k = 1, \ldots, N$ do
    Perform prediction update and compute $\hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}$ by
        $\hat{\mathbf{x}}_{k|k-1} = A\hat{\mathbf{x}}_{k-1|k-1}$
        $P_{k|k-1} = AP_{k-1|k-1}A^{\mathrm{T}} + Q$
    Perform measurement update and compute $\hat{\mathbf{x}}_{k|k}, P_{k|k}$ by
        $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k \left(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}\right)$
        $P_{k|k} = P_{k|k-1} - K_k S_{k|k-1} K_k^{\mathrm{T}}$
    where
        $\hat{\mathbf{y}}_{k|k-1} = C\hat{\mathbf{x}}_{k|k-1}$
        $S_{k|k-1} = CP_{k|k-1}C^{\mathrm{T}} + R$
        $K_k = P_{k|k-1}C^{\mathrm{T}}S_{k|k-1}^{-1}$
end for
```

Table 2.1: Terms Used in the Kalman Filter Equations

$\hat{\mathbf{x}}_{k\|k-1}$:	Predicted state
$P_{k\|k-1}$:	Covariance of the predicted state
$\hat{\mathbf{x}}_{k\|k}$:	Estimated state
$P_{k\|k}$:	Covariance of the estimated state
$\hat{\mathbf{y}}_{k\|k-1}$:	Predicted measurement
$\boldsymbol{\nu}_k \triangleq \mathbf{y}_k - \hat{\mathbf{y}}_{k\|k-1}$:	Measurement prediction error/ Innovation
$S_{k\|k-1}$:	Covariance of the innovation
K_k:	Kalman gain

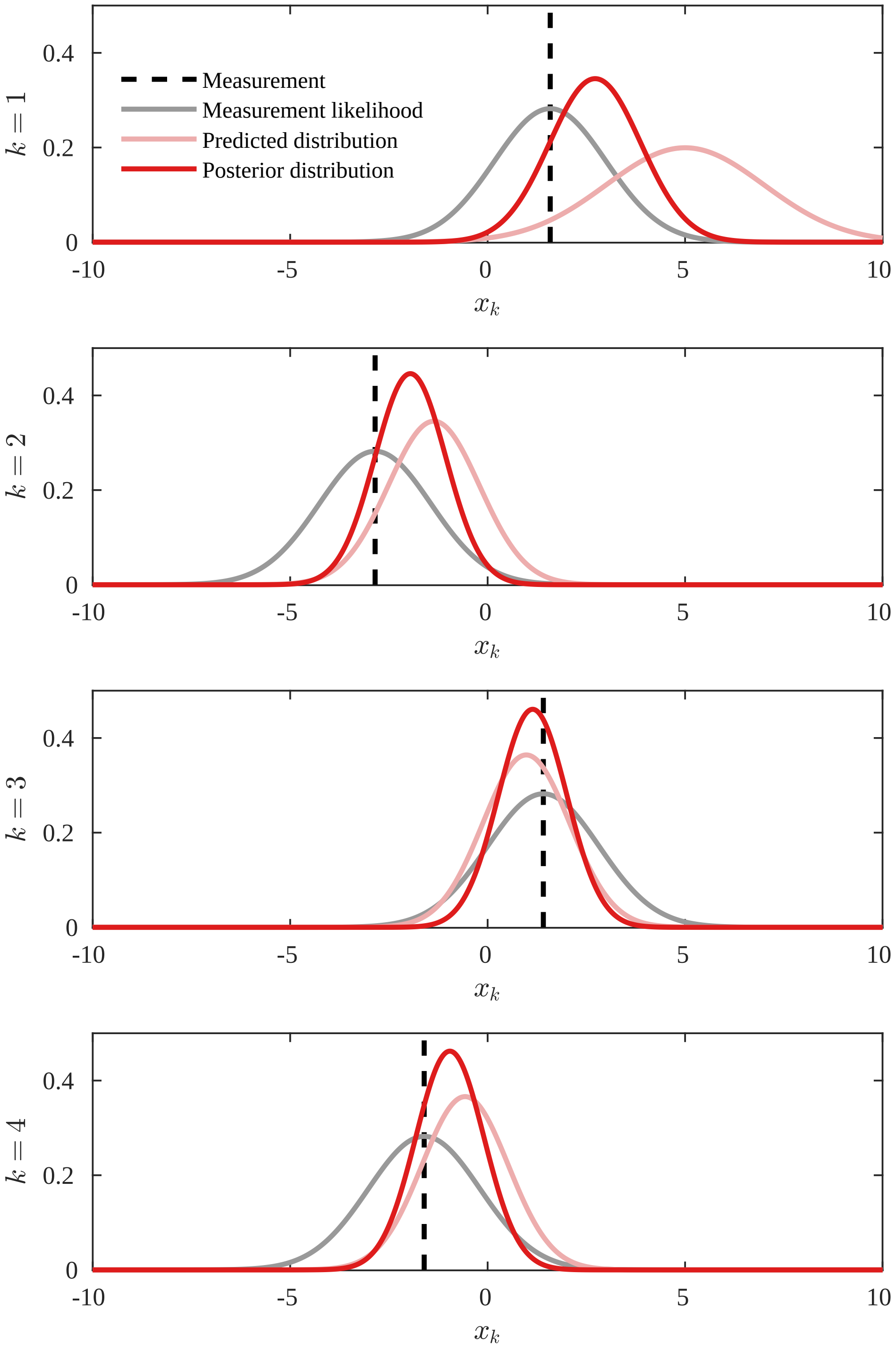


Figure 2.2: An illustration of the distributions obtained during four consecutive time steps of Bayesian filtering for the linear Gaussian system in (2.15).

and the observations of our model, respectively. The exposition in this section closely follows the presentation in [41, Ch. 10].

The marginal log-likelihood of all measurements $\log p(\mathbf{y})$ can analytically be decomposed as

$$\log p(\mathbf{y}) = \mathcal{L}(q(\mathbf{x})) + \mathrm{KL}(q(\mathbf{x}) \| p(\mathbf{x}|\mathbf{y})), \tag{2.18a}$$

where

$$\mathcal{L}(q(\mathbf{x})) = \int q(\mathbf{x}) \log \left[\frac{p(\mathbf{y}, \mathbf{x})}{q(\mathbf{x})} \right] \mathrm{d}\mathbf{x}, \tag{2.18b}$$

$$\mathrm{KL}(q(\mathbf{x}) \| p(\mathbf{x}|\mathbf{y})) = \int q(\mathbf{x}) \log \left[\frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})} \right] \mathrm{d}\mathbf{x}. \tag{2.18c}$$

$\mathrm{KL}(\cdot)$ denotes the Kullback-Leibler divergence between distributions $q(\mathbf{x})$ and $p(\mathbf{x}|\mathbf{y})$. As the KL divergence is guaranteed to be nonnegative, the following relation is valid for all $q(\cdot)$

$$\log p(\mathbf{y}) \geq \mathcal{L}(q(\mathbf{x})),$$

and hence $\mathcal{L}(\cdot)$ is interpreted to be a lower bound for the marginal log-likelihood.

Our aim is to minimize the KL divergence in (2.18), which is trivially solved by $q(\mathbf{x})$ being equal to $p(\mathbf{x}|\mathbf{y})$. However, this solution is not analytically tractable; therefore, we need to resort to an approximate solution by confining $q^*(\mathbf{x})$ to a family of functions for analytical convenience, i.e.,

$$q^*(\cdot) = \underset{q \,\in\, \mathcal{F}}{\mathrm{argmin}}\, \mathrm{KL}(q(\cdot) \| p(\cdot)) \; . \tag{2.19}$$

The challenge here is to specify a sufficiently rich and flexible family for the function of approximate distributions so that the result can approach to the true posterior while keeping the solution tractable. In this regard, the variational Bayesian inference is based on imposing a factorized form for $q(\cdot)$ as

$$q(\mathbf{x}) = \prod_{i=1}^{M} q_i(\mathbf{x}_i) \; . \tag{2.20}$$

This approximation is also referred to as the *mean field approximation* in the literature. In accordance with (2.18), to minimize the KL divergence, we can alternatively maximize the lower bound:

$$\begin{aligned} q^*(\mathbf{x}) &= \mathrm{argmin}\; \mathrm{KL}(q(\mathbf{x}) \| p(\mathbf{x} \mid \mathbf{y})) \\ &= \mathrm{argmax}\; \mathcal{L}(q(\mathbf{x})) \; . \end{aligned} \tag{2.21}$$

This is a variational optimization problem and iteratively maximized for each factor [41, Ch. 10] as in

$$\begin{aligned}
\mathcal{L}(q) &= \int \prod_i q_i\left(\mathbf{x}_i\right)\left\{\log p(\mathbf{y}, \mathbf{x}) - \sum_i \log q_i\left(\mathbf{x}_i\right)\right\} \mathrm{d}\mathbf{x} \\
&= \int q_j\left(\mathbf{x}_j\right)\left\{\int \log p(\mathbf{y}, \mathbf{x}) \prod_{i \neq j} q_i\left(\mathbf{x}_i\right) \mathrm{d}\mathbf{x}_i\right\} \mathrm{d}\mathbf{x}_j - \int q_j\left(\mathbf{x}_j\right) \log q_j\left(\mathbf{x}_j\right) \mathrm{d}\mathbf{x}_j + c \\
&= \int q_j\left(\mathbf{x}_j\right) \log \hat{p}\left(\mathbf{y}, \mathbf{x}_j\right) \mathrm{d}\mathbf{x}_j - \int q_j\left(\mathbf{x}_j\right) \log q_j\left(\mathbf{x}_j\right) \mathrm{d}\mathbf{x}_j + c \\
&= -\int q_j\left(\mathbf{x}_j\right) \frac{\log q_j\left(\mathbf{x}_j\right)}{\log \hat{p}\left(\mathbf{y}, \mathbf{x}_j\right)} \mathrm{d}\mathbf{x}_j + c, \qquad (2.22a)
\end{aligned}$$

where

$$\begin{aligned}
\log \hat{p}\left(\mathbf{y}, \mathbf{x}_j\right) &= \int \log p(\mathbf{y}, \mathbf{x}) \prod_{i \neq j} q_i \ \mathrm{d}\mathbf{x}_i \\
&= \mathbb{E}_{i \neq j}[\log p(\mathbf{y}, \mathbf{x})] + c\ . \qquad (2.22b)
\end{aligned}$$

Notice that the resulting expression in (2.22a) is the negative KL divergence between $q_j\left(\mathbf{x}_j\right)$ and $\hat{p}\left(\mathbf{y}, \mathbf{x}_j\right)$. Accordingly, the optimal distribution that maximizes the lower bound is obtained by minimizing the mentioned KL divergence by $q_j^*\left(\mathbf{x}_j\right) = \hat{p}\left(\mathbf{y}, \mathbf{x}_j\right)$. Consequently, we end up with the following expression

$$\log q_j^*\left(\mathbf{x}_j\right) = \mathbb{E}_{i \neq j}[\log p(\mathbf{y}, \mathbf{x})] + \mathrm{c}. \qquad (2.23)$$

The optimal distribution for the j^{th} factor is computed by

$$q_j^*\left(\mathbf{x}_j\right) = \frac{\exp\left(\mathbb{E}_{i \neq j}[\log p(\mathbf{y}, \mathbf{x})]\right)}{\int \exp\left(\mathbb{E}_{i \neq j}[\log p(\mathbf{y}, \mathbf{x})]\right) \mathrm{d}\mathbf{x}_j}\ . \qquad (2.24)$$

Each factor of the overall factorized distribution is computed by using the given equation while keeping the other factor fixed at its most recent estimate.

2.4 Gaussian Processes

A Gaussian process (GP) is essentially a collection of random variables, any finite subset of which have a joint Gaussian distribution, [42]. It can be interpreted as a generalization of multivariate Gaussian distribution over function space. A GP model defined for a scalar-valued function $f(\cdot)$ is denoted as

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')),$$

where *x* is the argument of the function. The GP is uniquely defined by its mean function $\mu(x)$ and its covariance function $k(x, x')$ given as

$$\mu(x) = \mathbb{E}[f(x)], \tag{2.25a}$$

$$k(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))]. \tag{2.25b}$$

By the definition of the GP, the function values evaluated at a finite number of inputs, $x_1, ..., x_N$, are normally distributed, i.e.,

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, K), \tag{2.26a}$$

where

$$\boldsymbol{\mu} = \begin{bmatrix} \mu(x_1) \\ \vdots \\ \mu(x_N) \end{bmatrix} \text{ and } K = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_N) \\ \vdots & & \vdots \\ k(x_N, x_1) & \dots & k(x_N, x_N) \end{bmatrix}. \tag{2.26b}$$

An example GP model and some functions sampled from this prior model are illustrated in Fig. 2.3. In particular, the model is specified by the zero-mean function and the squared exponential covariance function, i.e.,

$$\begin{aligned} f(x) \sim \mathcal{GP}(\mu(x), &k_{\text{SE}}(x, x')),\ x \in \mathbb{R}, \\ \text{where } \mu(x) &= 0, \\ k_{\text{SE}}(x, x') &= \sigma_f^2 \exp\left(-\frac{|x - x'|^2}{2l^2}\right), \end{aligned} \tag{2.27}$$

where the prior standard deviation is set to $\sigma_f = 4$, and the length scale is used as $l = 3$.

2.4.1 Gaussian Process Regression

The primary objective of utilizing a GP in stochastic modeling is typically to predict an unknown function $f(\cdot)$ at some query inputs regarding the collected observations.

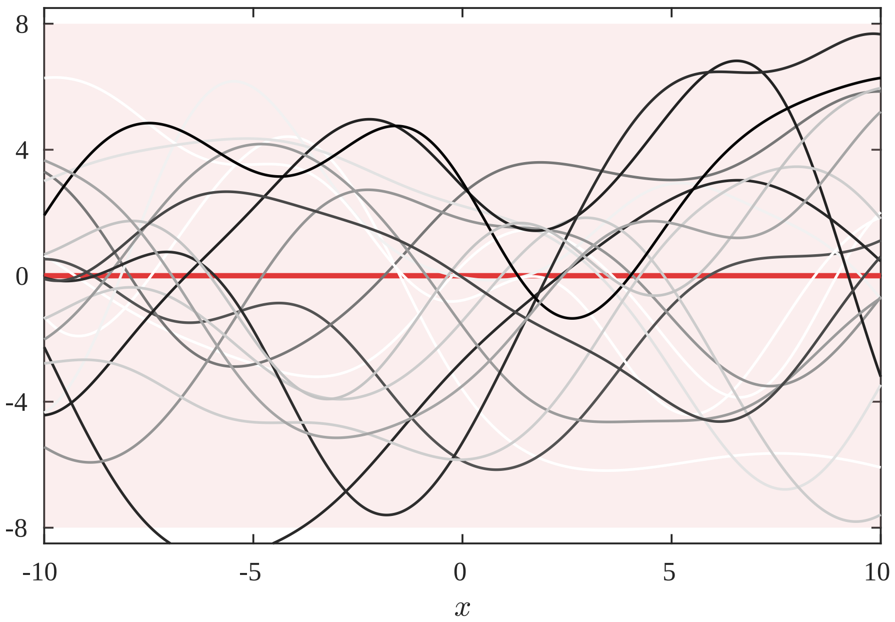


Figure 2.3: An illustration of the GP prior model in (2.27) with some sample functions. (The solid red line describes the mean function, which is identically zero; the shaded area indicates 2-σ confidence interval.)

In this pursuit, the GP provides analytical expressions for the unknown function values with well-calibrated uncertainty estimates.

Suppose we have a noisy observation y of the function $f(\cdot)$, which is described by the following measurement model,

$$y = f(x) + e, \;\; e \sim \mathcal{N}(0, \sigma_n^2), \tag{2.28}$$

where e denotes the independent sensor noise, which is Gaussian distributed. The aim is to estimate the values of the latent function, $\mathbf{f} \triangleq [f(x_1^{\mathbf{f}}) \; \dots \; f(x_{N^{\mathbf{f}}}^{\mathbf{f}})]^\top$ at the query inputs $\mathbf{x}^{\mathbf{f}} \triangleq [x_1^{\mathbf{f}} \; \dots \; x_{N^{\mathbf{f}}}^{\mathbf{f}}]^\top$. To this end, a set of measurements $\mathbf{y} \triangleq [y_1 \; \dots \; y_N]^\top$ are collected by observing the function at $\mathbf{x} \triangleq [x_1 \; \dots \; x_N]^\top$. In accordance with the GP model (2.26) and the measurement model (2.28), the joint distribution of the measurements and the function values can be written as

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(\mathbf{x}) \\ \boldsymbol{\mu}(\mathbf{x}^{\mathbf{f}}) \end{bmatrix}, \begin{bmatrix} K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I_N & K(\mathbf{x}, \mathbf{x}^{\mathbf{f}}) \\ K(\mathbf{x}^{\mathbf{f}}, \mathbf{x}) & K(\mathbf{x}^{\mathbf{f}}, \mathbf{x}^{\mathbf{f}}) \end{bmatrix} \right), \tag{2.29a}$$

where

$$\begin{aligned}\boldsymbol{\mu}(\mathbf{x}) &= [\mu(x_1)\ \dots\ \mu(x_N)]^\top,\\ \boldsymbol{\mu}(\mathbf{x}^\mathbf{f}) &= [\mu(x_1^\mathbf{f})\ \dots\ \mu(x_{N^\mathbf{f}}^\mathbf{f})]^\top,\\ K(\mathbf{x},\mathbf{x}^\mathbf{f}) &= \begin{bmatrix} k(x_1,x_1^\mathbf{f}) & \dots & k(x_1,x_{N^\mathbf{f}}^\mathbf{f})\\ \vdots & & \vdots\\ k(x_N,x_1^\mathbf{f}) & \dots & k(x_N,x_{N^\mathbf{f}}^\mathbf{f})\end{bmatrix}.\end{aligned} \tag{2.29b}$$

I_N indicates an N-by-N identity matrix. Consequently, we can analytically compute the conditional distribution $p(\mathbf{f}|\mathbf{y})$ as

$$p(\mathbf{f}|\mathbf{y}) = \mathcal{N}\left(\boldsymbol{\mu}^+, \Sigma^+\right)\ , \tag{2.30a}$$

where

$$\begin{aligned}\boldsymbol{\mu}^+ &= \boldsymbol{\mu}(\mathbf{x}^\mathbf{f}) + K\left(\mathbf{x}^\mathbf{f},\mathbf{x}\right)\left[K(\mathbf{x},\mathbf{x}) + \sigma_n^2 I_N\right]^{-1}(\mathbf{y} - \boldsymbol{\mu}(\mathbf{x}))\,,\\ \Sigma^+ &= K\left(\mathbf{x}^\mathbf{f},\mathbf{x}^\mathbf{f}\right) - K\left(\mathbf{x}^\mathbf{f},\mathbf{x}\right)\left[K(\mathbf{x},\mathbf{x}) + \sigma_n^2 I_N\right]^{-1} K\left(\mathbf{x},\mathbf{x}^\mathbf{f}\right).\end{aligned} \tag{2.30b}$$

If we select the underlying GP model to have zero mean, which is the common practice for many applications, the posterior mean reduces to a linear function of the observations as

$$\boldsymbol{\mu}^+ = K\left(\mathbf{x}^\mathbf{f},\mathbf{x}\right)\left[K(\mathbf{x},\mathbf{x}) + \sigma_n^2 I_N\right]^{-1}\mathbf{y}. \tag{2.31}$$

Accordingly, we can rewrite the posterior distribution in the following compact form,

$$p(\mathbf{f}|\mathbf{y}) \sim \mathcal{N}(A\mathbf{y}, P), \tag{2.32a}$$

where

$$A = K(\mathbf{x}^\mathbf{f},\mathbf{x})K_y^{-1}, \tag{2.32b}$$

$$P = K(\mathbf{x}^\mathbf{f},\mathbf{x}^\mathbf{f}) - K(\mathbf{x}^\mathbf{f},\mathbf{x})K_y^{-1}K(\mathbf{x},\mathbf{x}^\mathbf{f}), \tag{2.32c}$$

$$K_y = K(\mathbf{x},\mathbf{x}) + \sigma_n^2 I_N. \tag{2.32d}$$

As an illustration, the regression results of the prior model in (2.27) is given in Fig. 2.4.

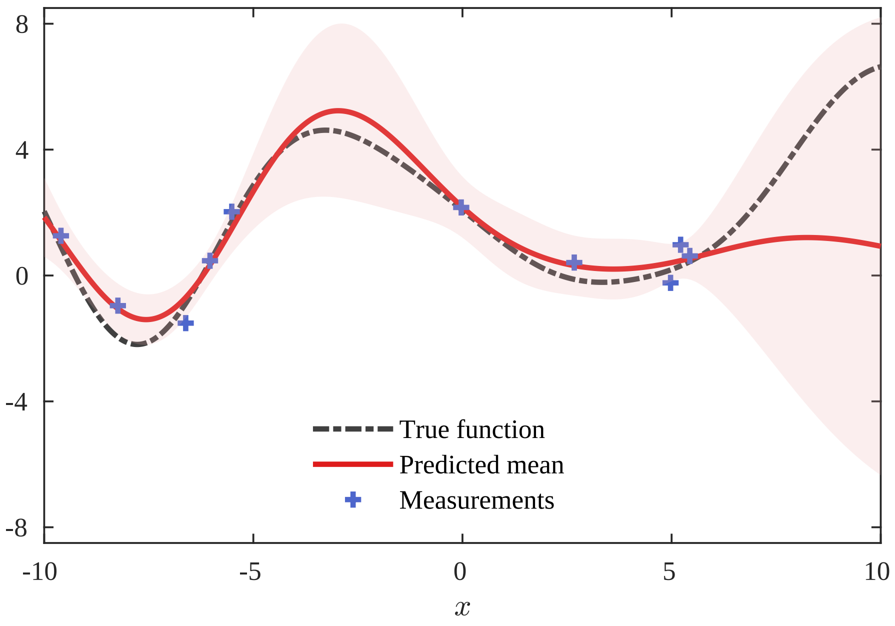


Figure 2.4: A GP regression example obtained by conditioning the prior model in (2.27) to the given observations. (The solid red line describes the posterior predictive mean; the shaded area indicates 2-σ confidence interval. The standard deviation of the measurement noise is set to $\sigma_n = 0.5$.)

2.4.2 Gaussian Process Approximations

Along with their favorable characteristics, such as robustness to overfitting and providing principled uncertainty information, GPs have been widely adopted mainly due to their flexibility in modeling. They do not rely on a parametric model, which is often too restrictive, to explain the observed data generation process. Instead, a GP model enables us to make predictions directly from observations. Accordingly, all available measurements are processed in a single batch by the standard GP regression formulae, as indicated in the previous section. Unfortunately, this attribute gives rise to the widely-known limitation of GPs that they scale with $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$ in computational and memory requirements, where n is the number of measurements. The specified complexities are computed for a naive implementation of the regression equations requiring the inversion and the storage of the Gram matrix, $[K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I_N]$.

Although it obviously depends on the accessible processing capabilities, the mentioned computational characteristics typically impede the utilization of GPs when the number of observations is more than a couple of thousands. Specifically, for the applications that deal with continuous streams of observations, e.g., target tracking, and system identification, it is not feasible to employ a GP model as the number of available measurements grows without a limit which imposes an obvious burden on naturally limited computational sources.

There is a well-established literature of studies that aim at improving the computational characteristics of GPs. The derivation of these *sparse* GP models relies on a broad spectrum of mechanisms. A group of methods simply downsamples the observations by selecting a subset of the overall dataset [43]. In another branch, the overall dataset is clustered into partitions, and local GP models are trained with each partition [44]. Then, the outputs of these multiple models are appropriately combined for prediction at a query point. Alternatively, a solid body of studies approximates the original GP by a set of *inducing variables* that is assumed to carry all necessary information about the prediction points [45, 46, 47]. A thorough review of the existing literature is outside the scope of this thesis work. Instead, in the remaining of this section, we introduce the methods adopted in the formulation of our extended object trackers in the following chapters.

2.4.2.1 Sparse Gaussian Processes using Pseudo-inputs

In their seminal paper, [45], Snelson and Ghahramani offer to form a *pseudo dataset* that can *summarize* the original dataset adequately with fewer number of data points, which in turn scales down the computational complexity of GP training and regression.

The GP model offers the following posterior predictive density for an observation y^* at a query input x^*

$$p(y^*|x^*, \mathcal{D}, \boldsymbol{\theta}) = \mathcal{N}\left(y^*; \mathbf{k}_{\mathbf{x}*}^\top \left(K_N + \sigma^2 I\right)^{-1} \mathbf{y}, k_{**} - \mathbf{k}_{\mathbf{x}*}^\top \left(K_N + \sigma^2 I\right)^{-1} \mathbf{k}_{\mathbf{x}*} + \sigma^2\right), \tag{2.33}$$

where $\mathcal{D}$ indicates the complete set of available observations, which consists of the measurements $\mathbf{y} = \{y_i\}_{i=1}^N$ obtained at the corresponding inputs $\mathbf{x} = \{x_i\}_{i=1}^N$, i.e.,

$\mathcal{D} = \{\mathbf{y}, \mathbf{x}\}$; $\boldsymbol{\theta}$ denotes the parameters of the covariance function, and σ is the measurement noise standard deviation. The indices of the utilized covariance matrices are specified as follows

$$\begin{aligned} [\mathbf{k}_{\mathbf{x}*}]_i &= k\left(x_i, x^*\right), \\ [K_N]_{i,j} &= k(x_i, x_j), \\ k_{**} &= k(x^*, x^*), \end{aligned} \tag{2.34}$$

where $k(\cdot, \cdot)$ is the covariance function.

Both the predicted mean and variance in (2.33) expectedly depend on the query input x^*, as it essentially relates y^* to the available observations. Furthermore, the predicted statistics are determined by the dataset $\mathcal{D}$ itself. SPGP suggests to substitute a pseudo dataset $\bar{\mathcal{D}}$ instead of the original one while regarding the predicted distribution in (2.33). In this pursuit, the idea is to construct a comparatively smaller dataset, which consists of M observations.

The pseudo dataset is defined to compose of $\overline{\mathbf{x}} = \{\overline{x}_i\}_{i=1}^{M}$ and $\overline{\mathbf{f}} = \left\{\overline{f}_i\right\}_{i=1}^{M}$, which are the inputs and pseudo observations/targets, respectively. Assuming that we are given the pseudo dataset, then the corresponding posterior predictive density is computed as

$$p(y^*|x^*, \overline{\mathbf{x}}, \overline{\mathbf{f}}) = \mathcal{N}\left(y^*; \mathbf{k}_{\overline{\mathbf{x}}*}^{\top} K_M^{-1} \overline{\mathbf{f}}, k_{**} - \mathbf{k}_{\overline{\mathbf{x}}*}^{\top} K_M^{-1} \mathbf{k}_{\overline{\mathbf{x}}*} + \sigma^2\right), \tag{2.35a}$$

where

$$\begin{aligned} [\mathbf{k}_{\overline{\mathbf{x}}*}]_i &= k\left(\overline{x}_i, x^*\right), \\ [K_M]_{i,j} &= k\left(\overline{x}_i, \overline{x}_j\right). \end{aligned} \tag{2.35b}$$

Additionally, in accordance with the pseudo dataset, the likelihood of the original dataset can be written as

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}, \overline{\mathbf{x}}, \overline{\mathbf{f}}) &\approx \prod_{n=1}^{N} p\left(y_n | x_n, \overline{\mathbf{x}}, \overline{\mathbf{f}}\right) \\ &= \mathcal{N}\left(\mathbf{y}; K_{NM} K_M^{-1} \overline{\mathbf{f}}, \Lambda + \sigma^2 I\right), \end{aligned} \tag{2.36a}$$

where

$$\begin{aligned}
\Lambda &= \mathrm{diag}(\boldsymbol{\lambda}) \ , \\
[\boldsymbol{\lambda}]_n &= k_{nn} - \mathbf{k}_{\overline{\mathbf{x}}n}^\top K_M^{-1} \mathbf{k}_{\overline{\mathbf{x}}n} \ , \\
k_{nn} &= k(x_n, x_n) \ , \\
[\mathbf{k}_{\overline{\mathbf{x}}n}]_i &= k(\overline{x}_i, x_n) \ , \\
[K_{NM}]_{i,j} &= k\left(x_i, \overline{x}_j\right) \ .
\end{aligned} \tag{2.36b}$$

It is important to note that the factorized expression in (2.36a) is based on the assumption that each measurement y_n is independent from the remaining measurements conditioned on the psuedo dataset. In other words, the pseudo dataset is selected in such a way that it provides the sufficient statistics for the real observations.

Subsequently, it is intuitive to impose a Gaussian prior on the pseudo targets that complies with the underlying GP model, i.e., $p(\overline{\mathbf{f}}|\overline{\mathbf{x}}) = \mathcal{N}\left(\overline{\mathbf{f}}; \mathbf{0}, K_M\right)$. Then, one can analytically compute the posterior distribution of the targets,

$$p(\overline{\mathbf{f}}|\mathcal{D}, \overline{\mathbf{x}}) = \mathcal{N}\left(\overline{\mathbf{f}}; K_M Q_M^{-1} K_{MN} \left(\Lambda + \sigma^2 I\right)^{-1} \mathbf{y}, K_M Q_M^{-1} K_M\right), \tag{2.37a}$$

where

$$Q_M = K_M + K_{MN} \left(\Lambda + \sigma^2 I\right)^{-1} K_{NM}. \tag{2.37b}$$

Accordingly, the predictive distribution at the query point transforms into

$$\begin{aligned}
p\left(y^* | x^*, \mathcal{D}, \overline{\mathbf{x}}\right) &= \int \mathrm{d}\overline{\mathbf{f}} p\left(y^* | \mathbf{x}^*, \overline{\mathbf{x}}, \overline{\mathbf{f}}\right) p(\overline{\mathbf{f}}|\mathcal{D}, \overline{\mathbf{x}}) \\
&= \mathcal{N}\left(y_*; \mu_*, \sigma_*^2\right) ,
\end{aligned} \tag{2.38a}$$

where

$$\begin{aligned}
\mu_* &= \mathbf{k}_{\overline{\mathbf{x}}*}^\top Q_M^{-1} K_{MN} \left(\Lambda + \sigma^2 I\right)^{-1} \mathbf{y}, \\
\sigma_*^2 &= k_{**} - \mathbf{k}_{\overline{\mathbf{x}}*}^\top \left(K_M^{-1} - Q_M^{-1}\right) \mathbf{k}_{\overline{\mathbf{x}}*} + \sigma^2.
\end{aligned} \tag{2.38b}$$

Notice that the expression in (2.38a) again follows from the assumption that y^* is independent from the original observations conditioned on the pseudo dataset, i.e.,

$$p\left(y^* | \mathbf{x}^*, \mathcal{D}, \overline{\mathbf{x}}, \overline{\mathbf{f}}\right) = p\left(y^* | \mathbf{x}^*, \overline{\mathbf{x}}, \overline{\mathbf{f}}\right) .$$

In this formulation, the inversion of $(\Lambda + \sigma^2 I)$ scales with only $\mathcal{O}(n)$ as it is a diagonal matrix. By using some precomputations, the computational complexity of prediction per test case reduces to $\mathcal{O}(M^2)$, [45].

The study also investigates the training process of the suggested approximate GP model, where the aim is to optimize the pseudo-input locations, $\overline{\mathbf{x}}$, together with the hyperparameters, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \sigma^2\}$. This is achieved by maximizing the following marginal likelihood with respect to the parameters by an iterative gradient-based approach.

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}, \overline{\mathbf{x}}, \boldsymbol{\Theta}) &= \int \mathrm{d}\overline{\mathbf{f}} p(\mathbf{y}|\mathbf{x}, \overline{\mathbf{x}}, \overline{\mathbf{f}}) p(\overline{\mathbf{f}}|\overline{\mathbf{x}}) \\ &= \mathcal{N}\left(\mathbf{y}; \mathbf{0}, K_{NM} K_M^{-1} K_{MN} + \boldsymbol{\Lambda} + \sigma^2 I\right) \end{aligned} \tag{2.39}$$

The major shortcomings of presented approach can be listed as follows:

- To determine the locations of the pseudo-inputs, all available measurements are to be processed to maximize the likelihood given in (2.39). However, it is not practical for an online problem as the number measurements will sequentially grow in time without a limit.
- The algorithm relies on a gradient-based optimization scheme, and it does not present a closed form solution for the locations of the pseudo inputs. This might be problematic for applications with strict real-time requirements to fulfill.

2.4.2.2 Recursive Gaussian Processes

In the previous section, a GP approximation based on substituting the original dataset with a smaller pseudo dataset is introduced. This method inherently assumes that the observations are collected and stored prior to forming the sparse GP model. On the other hand, this assumption is violated in many practical applications, where the *so-called* streaming data becomes sequentially available. In this setting, it is not practically feasible to store all available information and process them to obtain an approximate model; instead, there is a need for dedicated solutions that is able to incrementally update the sparse GP approximation processing only the new data. These issues are addressed by the studies [48] and [49] under the name of recursive Gaussian processes.

In this approach, the objective is to update the posterior distribution of the latent function values $\overline{\mathbf{f}} = \{\overline{f}_i\}_{i=1}^{M}$ at the inputs $\overline{\mathbf{x}} = \{\overline{x}_i\}_{i=1}^{M}$, in a recursive manner. $\overline{\mathbf{f}}$ and $\overline{\mathbf{x}}$ are called the inducing targets and the inducing inputs, respectively. $\mathbf{y} = \{y_i\}_{i=1}^{N}$ indicates the overall set of observations generated at the inputs $\mathbf{x} = \{x_i\}_{i=1}^{N}$. The posterior $p(\overline{\mathbf{f}}|\mathbf{y})$ is first expanded as the collection of the following terms by applying the Bayes' rule iteratively,

$$p(\overline{\mathbf{f}}|y_{1:N}) \propto p(y_N|\overline{\mathbf{f}}, y_{1:N-1})p(\overline{\mathbf{f}}|y_{1:N-1}), \tag{2.40a}$$

$$\propto \ldots \underbrace{p(y_k|\overline{\mathbf{f}}, y_{1:k-1}) \ldots p(\overline{\mathbf{f}})}_{p(\overline{\mathbf{f}}|y_{1:k})}. \tag{2.40b}$$

Subsequently, $\overline{\mathbf{f}}$ is assumed to provide the sufficient statistics for y_k. With this assumption, y_k becomes conditionally independent of all the other measurements $y_{1:k-1}$ given $\overline{\mathbf{f}}$, i.e.,

$$p(y_k|\overline{\mathbf{f}}, y_{1:k-1}) \approx p(y_k|\overline{\mathbf{f}}). \tag{2.41}$$

Notice that this is exactly the same approximation which provides the basis of the approach in [45]. In this regard, the inducing inputs, $\overline{\mathbf{x}}$, together with the corresponding function values, $\overline{\mathbf{f}}$, can directly be interpreted as the pseudo dataset. It is also important to note that this formulation assumes that the inducing inputs are fixed and not treated in a Bayesian framework, as such an attempt often leads to an analytically intractable model, [45].

Thereafter, the derivation follows a different path from [45] with the aim of employing the standard Kalman filtering techniques for inference. In particular, the above assumption leads to a setting where we essentially treat $\overline{\mathbf{f}}$ to be the latent variable and the measurements provide noisy observations of it. Accordingly, once the measurement likelihood and the initial prior densities are defined, it is possible to apply recursive Bayesian inference for $\overline{\mathbf{f}}$. To this end, the relation between each measurement, y_k, and $\overline{\mathbf{f}}$ is provided by the underlying the GP model as follows.

$$\begin{bmatrix} y_k \\ \overline{\mathbf{f}} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(x_k, x_k) + R & K(x_k, \overline{\mathbf{x}}) \\ K(\overline{\mathbf{x}}, x_k) & K(\overline{\mathbf{x}}, \overline{\mathbf{x}}) \end{bmatrix}\right) \tag{2.42}$$

Accordingly, we can compute the likelihood and prior densities as

$$p(y_k|\bar{\mathbf{f}}) = \mathcal{N}(y_k; H_k\bar{\mathbf{f}}, R_k), \tag{2.43a}$$

$$p(\bar{\mathbf{f}}) = \mathcal{N}(\mathbf{0}, P_0), \tag{2.43b}$$

where

$$H_k = K(x_k, \bar{\mathbf{x}})K(\bar{\mathbf{x}}, \bar{\mathbf{x}})^{-1}, \tag{2.43c}$$

$$R_k = k(x_k, x_k) + R - K(x_k, \bar{\mathbf{x}})K(\bar{\mathbf{x}}, \bar{\mathbf{x}})^{-1}K(\bar{\mathbf{x}}, x_k), \tag{2.43d}$$

$$P_0 = K(\bar{\mathbf{x}}, \bar{\mathbf{x}}). \tag{2.43e}$$

We can immediately construct a linear Gaussian model that implies the same probabilistic characteristics for y_k and $\bar{\mathbf{f}}$ as in (2.43), which in turn will be used to infer the latent function values recursively by a standard Kalman filter,

$$\bar{\mathbf{f}}_{k+1} = \bar{\mathbf{f}}_k, \tag{2.44a}$$

$$y_k = H_k\,\bar{\mathbf{f}}_k + e_k, \quad e_k \sim \mathcal{N}(0, R_k), \tag{2.44b}$$

$$\bar{\mathbf{f}}_0 \sim \mathcal{N}(0, P_0). \tag{2.44c}$$

The regression results obtained by the recursive GP approach for the model in (2.27) are shown in Fig. 2.5.

2.4.2.3 Reduced-Rank Gaussian Processes

In this section, we present another approach to approximate a GP model that brings in favorable computational properties and an intuitive interpretation. The method takes a different perspective and relies on the spectral domain representation of GPs. Moreover, the resulting method facilitates a recursive implementation, which naturally enables dealing with streaming data.

In [50], Solin and Särkkä introduce an approximation to the GP modeling, which solves the eigendecomposition problem of the Laplace operator, ∇^2, by using a boundary condition on a confined domain Ω, i.e.,

$$\begin{cases} -\nabla^2\phi_j(x) = \lambda_j^2\phi_j(x), & x \in \Omega \\ \phi_j(x) = 0, & x \in \partial\Omega. \end{cases} \tag{2.45}$$

The approximation then relies on the *Wiener-Khintchin theorem* which states the Fourier duality of the covariance and the power spectral density functions for a stationary covariance function, i.e., $k(r) \triangleq k(x, x')$ where $r \triangleq x - x'$,

$$\begin{aligned} k(r) &= \frac{1}{(2\pi)^d} \int S(\omega) e^{\mathrm{i}\omega^\top r} \mathrm{d}\omega, \\ S(\omega) &= \int k(r) e^{-\mathrm{i}\omega^\top r} \mathrm{d}r. \end{aligned} \tag{2.46}$$

Examples of stationary covariance functions include the squared exponential and the Matérn class, which are arguably the most widely used functions in the GP literature. The definitions of these covariance functions and their spectral densities are given in (2.47) and (2.48), respectively.

$$k_{\mathrm{SE}}(r) = \sigma_f^2 \exp\left(-\frac{r^2}{2l^2}\right), \tag{2.47a}$$

$$k_{\mathrm{M}}(r) = \sigma_f^2 \frac{2^{1-v}}{\Gamma(v)} \left(\frac{\sqrt{2v}r}{l}\right)^v K_v\left(\frac{\sqrt{2v}r}{l}\right) \tag{2.47b}$$

$$S_{\mathrm{SE}}(s) = \sigma_f^2 \sqrt{2\pi l^2} \exp\left(-\frac{\pi^2 l^2 s^2}{2}\right) \tag{2.48a}$$

$$S_{\mathrm{M}}(s) = \sigma_f^2 \frac{2\pi^{\frac{1}{2}} \Gamma\left(v + \frac{1}{2}\right) (2v)^v}{\Gamma(v) l^{2v}} \left(\frac{2v}{l^2} + s^2\right)^{-\left(v + \frac{1}{2}\right)} \tag{2.48b}$$

In [50], an approximate expression for the stationary covariance function k_θ is derived as follows,

$$k_\theta\left(x, x'\right) \approx \sum_j S_\theta(\sqrt{\lambda_j}) \phi^{(j)}(x) \phi^{(j)}\left(x'\right), \tag{2.49}$$

where $S_\theta(\cdot)$ is the spectral density function, λ_j and $\phi^{(j)}(\cdot)$ are the j^{th} eigenvalue and eigenfunction of the Laplace operator solved in the specified domain, and θ denotes the hyperparameters of the covariance function.

One of the appealing properties of the decomposition in (2.49) is that the hyperparameters affect the expansion exclusively through the spectral density, while the eigenfunctions depend solely on the choice of the domain.

Let us consider an example to form an approximate GP model of a latent function $f(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^2$. Without imposing any prior information, we choose the domain of interest to be a rectangular-shaped compact subset, i.e., $\Omega = [-L_1, L_1] \times [-L_2, L_2]$, for which the eigenvalues are eigenfunctions are available with the following analytic expressions,

$$\phi^{(j_1,j_2)}(\mathbf{x}) = \prod_{d=1}^{2} \frac{1}{\sqrt{L_d}} \sin\left(\frac{\pi j_d \left(x_d + L_d\right)}{2L_d}\right), \tag{2.50}$$

$$\lambda^{(j_1,j_2)} = \sum_{k=1}^{n_x} \left(\frac{\pi j_k}{2L_k}\right)^2, \tag{2.51}$$

where x_d is the d^{th} component of the input vector $\mathbf{x}$, and (j_1, j_2) is an index variable. The first 16 basis functions are illustrated in Fig. 2.6 for the given domain.

The basic idea of the reduced-rank approximation is to truncate the expansion in (2.49) to a finite number of functions, i.e.,

$$k\left(x, x'\right) = \sum_{j=1}^{m} S(\sqrt{\lambda_j})\phi^{(j)}(x)\phi^{(j)}\left(x'\right). \tag{2.52}$$

This is expected to be a good approximation as the eigenvalues of the Laplace operator is monotonically increasing and the spectral density goes to zero fast with increasing frequencies for bounded covariance functions, [50]. The interested readers can refer to [50] for detailed analysis and convergence proofs.

The truncated covariance function in (2.52) implies an equivalent linear model [42, Ch. 2.2],

$$f(x) \sim \mathcal{GP}\left(0, k(x, x')\right) \;\Leftrightarrow\; f(x) = \sum_{j=1}^{m} f^{(j)}\phi^{(j)}(x), \quad \text{where } f^{(j)} \sim \mathcal{N}\left(0, S(\sqrt{\lambda_j})\right). \tag{2.53}$$

This model suggests that the eigenfunctions of the Laplace operator $\{\phi^{(j)}(\cdot)\}_{j=1}^{m}$ actually form a set of basis functions to describe our model, and the unknown weights $\{f^{(j)}\}_{j=1}^{m}$ have a Gaussian prior distribution.

We can reorganize (2.53) as

$$f(x) = \Phi(x)^\top \mathbf{f}, \text{ where } \mathbf{f} \sim \mathcal{N}\left(\mathbf{0}, S\right), \tag{2.54a}$$

where

$$\Phi(x) \triangleq [\phi^{(1)}(x) \dots \phi^{(M)}(x)]^\top , \tag{2.54b}$$

$$\mathbf{f} \triangleq [f^{(1)} \dots f^{(M)}]^\top , \tag{2.54c}$$

$$S = \text{diag}\left(S(\sqrt{\lambda_1}), \dots, S(\sqrt{\lambda_M})\right) . \tag{2.54d}$$

This expression concludes the derivation of the reduced-rank approximation, where the original GP model is transformed into a linear model with a latent variable $\mathbf{f}$.

According to (2.54), learning the latent function $f(\cdot)$ from a set of noisy observations $\{y_k\}_{k=1}^N$ amounts to computing the posterior distribution $p(\mathbf{f}|y_{1:N})$. Let us define a generic observation model,

$$\begin{aligned} y &= f(x) + e \\ &= \Phi(x)^\top \mathbf{f} + e, \end{aligned} \tag{2.55}$$

where e stands for the independent sensor noise. Notice that this formulation enables a Bayesian treatment of the problem, and we can easily compute the posterior distribution of $\mathbf{f}$ by employing one of the well-studied inference methods in the literature. In this regard, the resulting formulation fulfills the requirement to provide an online solution while the measurements become sequentially available in time. In particular, one can apply the Bayes' rule iteratively to expand the posterior as

$$\begin{aligned} p(\mathbf{f}|y_{1:N}) &\propto p(y_N|\mathbf{f}, y_{1:N-1})\, p(\mathbf{f}|y_{1:N-1}) \\ &= p(y_N|\mathbf{f})\, p(\mathbf{f}|y_{1:N-1}), \end{aligned} \tag{2.56}$$

where second equation follows from the fact that $\mathbf{f}$ provides the sufficient statistics for y_k, i.e., $p(y_k|\mathbf{f}, y_{1:k-1}) = p(y_k|\mathbf{f})$ as implied by (2.55). Additionally, for the specific case where the sensor noise is Gaussian, the exact solution is accessible by a measurement update of the Kalman filter. The regression results obtained by the reduced-rank GP approach for the model in (2.27) are shown in Fig. 2.7.

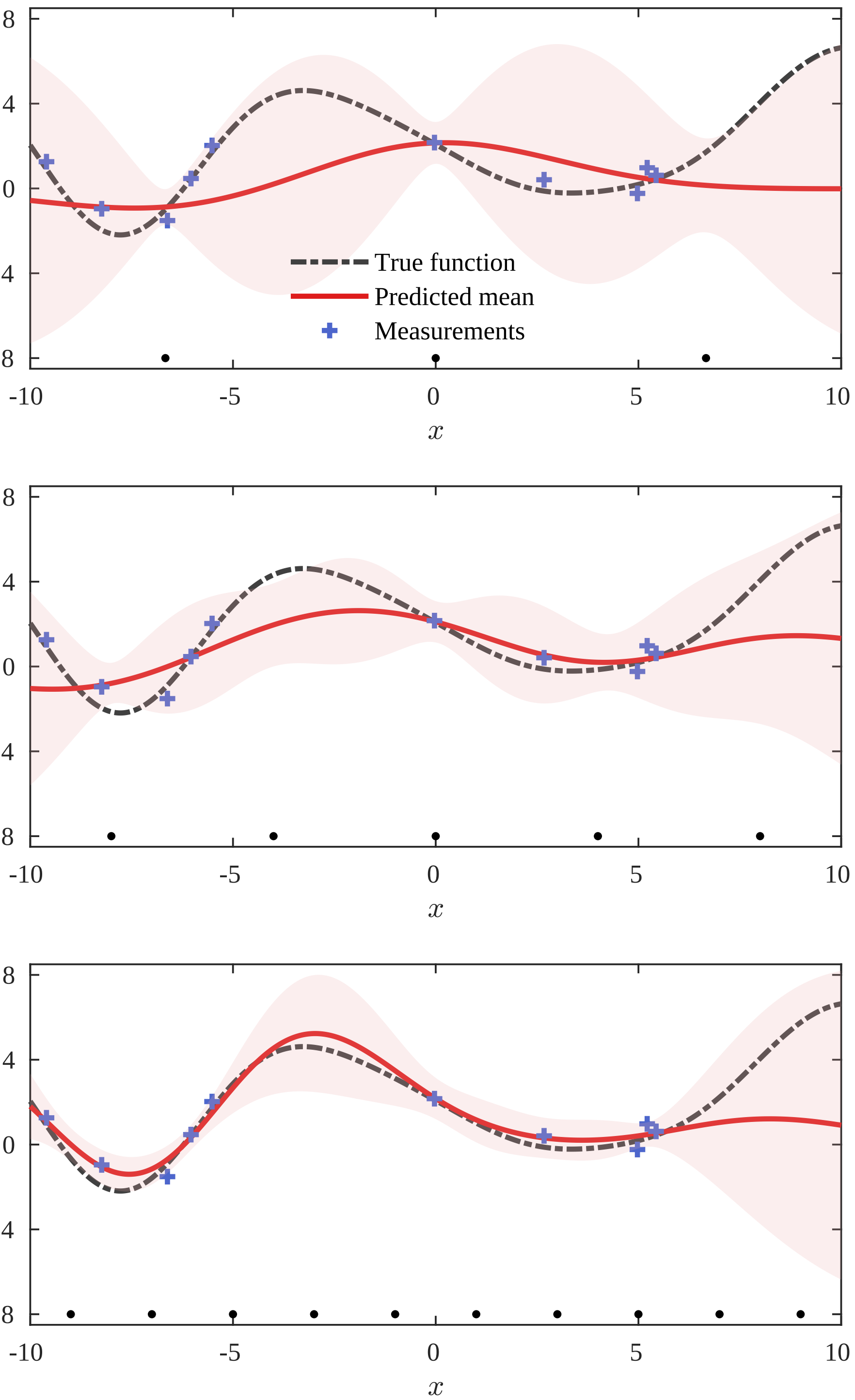


Figure 2.5: Regression results obtained by the recursive GP approach for the model in (2.27). The number of inducing inputs are selected as 3, 5, 10 in top, middle and bottom figures, respectively. (The solid red line describes the posterior predictive mean; the shaded area indicates 2-σ confidence interval. The black dots over the x-axis represent the locations of the basis points.)

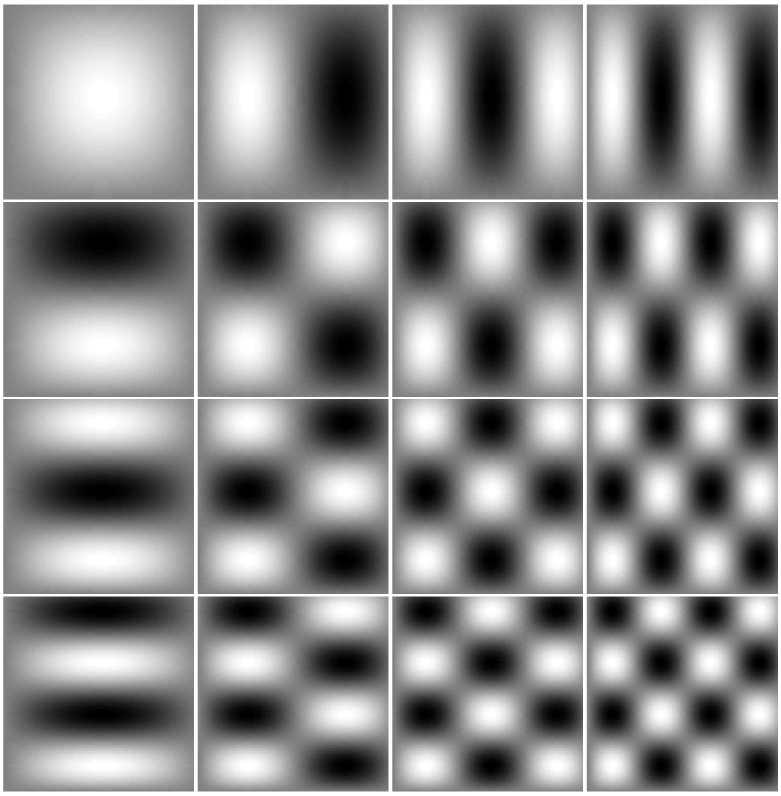


Figure 2.6: The first 16 eigenfunctions for the rectangular domain. The largest (+1) and smallest (-1) values of the functions are mapped to solid white and black, respectively; the values in between are plotted according to a linear interpolation.

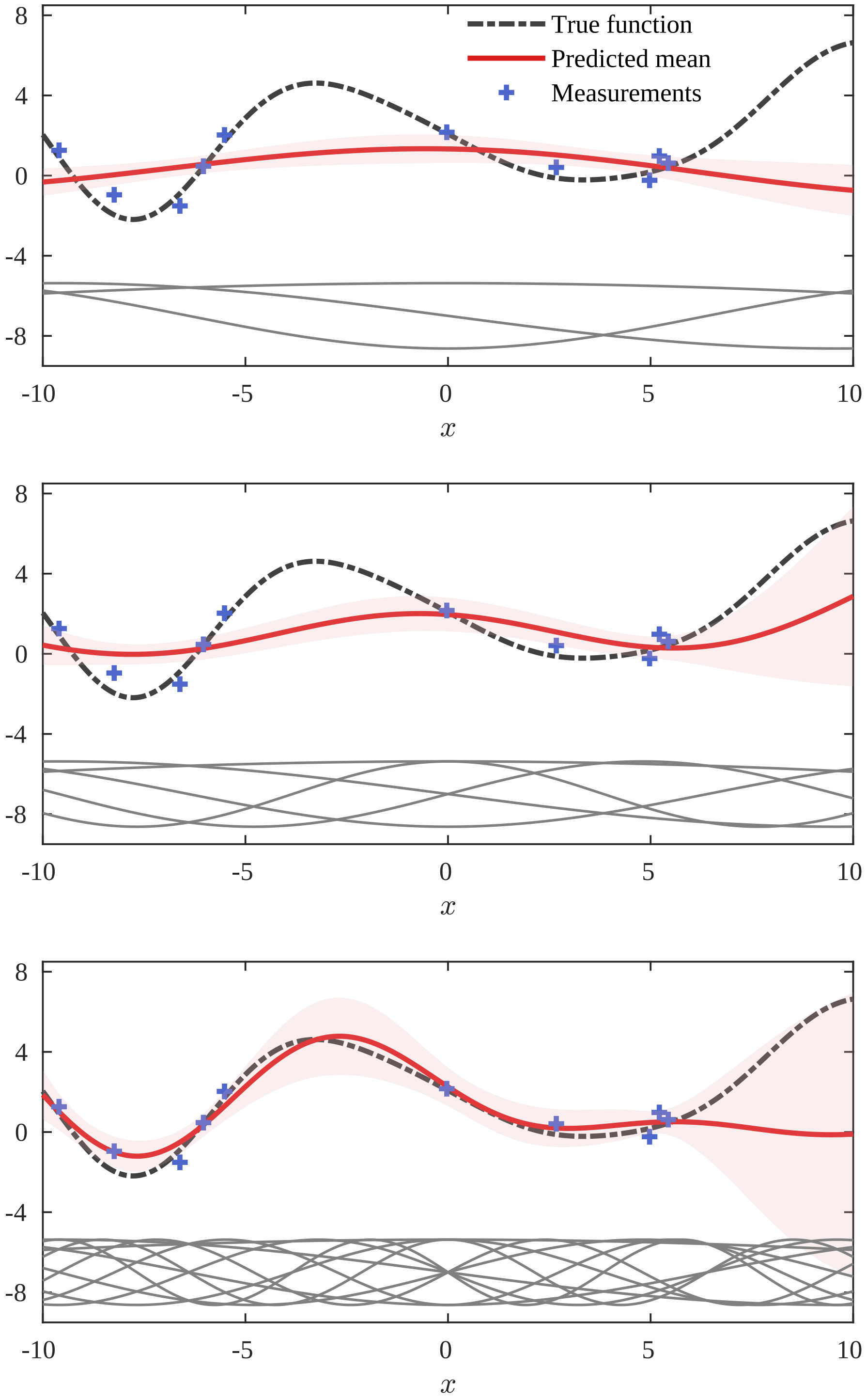


Figure 2.7: Regression results obtained by the reduced-rank GP approach for the model in (2.27). The number of basis functions are selected as 3, 5, 10 in top, middle and bottom figures, respectively. (The solid red line describes the posterior predictive mean; the shaded area indicates 2-σ confidence interval. The basis functions used in each case are illustrated over the x-axis.)

CHAPTER 3

THREE-DIMENSIONAL EXTENDED OBJECT TRACKING USING GAUSSIAN PROCESSES

3.1 Introduction

Object tracking can be described as making inference about the unknown kinematic properties of an object using sequentially available sensor data. The problem is explicitly referred to as *point object tracking* when the object of interest is relatively small that it may occupy a single cell of the sensor device. In this case, the number of measurements returned from the object is limited to be at most one per scan. On the other hand, it is dubbed *extended object tracking* (EOT) when the resolution of the sensor is sufficiently high relative to the size of the object so that it may be detected by several cells of the sensor. Thus, the object potentially originates multiple measurements at a single scan in this setting.

There is a solid body of literature considering the EOT problem, specifically in two-dimensional (2D) space (see [3] for a comprehensive survey). The methods of this category process 2D measurements by relying on various extent models. These extent representations exhibit significant variance in their compactness and expressive power. For example, a group of EOT algorithms imposes simple geometric shape models such as a circle, a rectangle, or an ellipse, [51, 5, 8, 9, 10, 11, 52]. These essentially achieve extent modeling with only a few parameters at the cost of limited potential for shape description. In another line of research, random hyper-surface models (RHM) suggest a more flexible extent representation for star-convex objects, [12], [13]. The model is based on the Fourier series expansion of the spatial extent, and the coefficients of the expansion are estimated together with the kinematics.

Similarly, the latent extent of a star-convex object is described by a Gaussian process (GP) in [26] and [25]. With its favorable analytical properties and close connections to the Bayesian paradigm, this model enables effective estimation of the object pose and its shape. Several adaptations of this approach have been investigated for various application settings, [27, 53, 29, 30, 54, 31, 28, 55].

On the other hand, there have been remarkably few attempts to tackle the EOT problem in three-dimensional (3D) space, despite the increasing accessibility of sensors such as depth cameras and LIDARs generating 3D data. Furthermore, the majority of these attempts are direct generalizations of the basic geometric models and represent the extent via, for example, an ellipsoid or a bounding box. One of the few exceptions that aim at a more generic extent description is proposed in [15]. The method assumes that 3D object surface can be constructed by some transformations, e.g., translation, rotation, of a plane curve. Then, it adopts the notion of RHM to derive a general tracking framework. This approach necessitates a special formulation of the recursive estimator in accordance with the particular transformation considered. However, the required prior information about the object shape may not be available for a standard tracking application.

Another branch of studies avoids imposing a parametric shape description; instead it is based on a point cloud representation of the shape, which is formed by collecting measurements over time, [21, 22, 23]. These methods facilitate joint tracking and shape learning of arbitrary objects in the presence of continuously available, high-precision and informative measurements. However, as they do not feature a principled representation of the underlying shape, there arise robustness issues with the sparsity of the measurements, for example, due to increasing distance, change of the vantage point and occlusions. In addition, the storage and the computational requirements scale with the size of the object extent.

In this chapter, we consider the problem of tracking 3D objects while simultaneously learning their shapes using 3D point cloud data. 3D measurements carry substantial information such that they not only convey clues about the kinematics but also reveal characteristics of the object extent naturally. However, estimating unknown shapes from noisy point cloud data is a challenging task, and the problem gets even more severe when the objects are in motion. This is mainly due to the inherent interdependence between the pose and the shape description. Therefore, the reliable estimation of one necessitates precise information about the other. With this in mind, we formulate the problem as the joint estimation of both kinematic and shape variables in a unified framework. For the description of the object extent, we adopt a GP-based approach that facilitates a flexible representation with favorable analytical properties. In particular, the contributions can be listed as follows.

- We propose two novel probabilistic representations for 3D extent. The first one expresses the unknown 3D surface by a radial function in spherical coordinates. The second exploits the correspondence between a 3D shape and its projections onto multiple planes, and thus characterizes the original 3D surface by a collection of 2D contours of these projections. The probabilistic shape representation is achieved by modeling the above descriptions by GPs without imposing any parametric form. By doing so, we attain a flexible basis to estimate the shape of a wide range of objects.

- This approach enables us to treat the unknown extent within the Bayesian framework such that the posterior distribution offers an analytical expression for the object shape with well-defined confidence intervals, and any available prior information can be incorporated easily.

- We develop measurement models to express the relation between the point measurements and the object extent using an efficient approximation of the GP regression.

- The kinematics and the object extent are efficiently inferred by an extended Kalman filter regarding a unified state-space model.

- The orientation of the object is described by the unit quaternions, and a novel rotational motion model is derived for effective estimation of the full 3D orientation and the angular rates.

- The performance of the suggested algorithms is comprehensively evaluated using both simulated and real measurements.

3.2 Extent Model for 3D Objects

Different types of sensors provide different level of details about objects of interest. One particular group of sensors, such as depth cameras and LIDARs, generates 3D point cloud data that capture salient characteristics of their environment. It is an open research question as to how we can harness the full potential of the available information. Other than its potential benefits in tracking such as clutter rejection and data association; a detailed shape estimate can provide valuable insight for the classification of the object, which in turn may prove useful to anticipate future behavior. With this improved perception capabilities, it might be possible to develop tailored ways of interaction with the environment.

To accomplish effective shape learning by processing 3D point cloud data, a suitable description of the object extent needs to be formulated that meets the following specifications. First, it is required to have high representational power to be able to apply to a wide range of objects with various 3D shapes. In addition, it should be sufficiently compact so that it will enable an efficient online tracking algorithm.

In this regard, we model the object shape in spherical coordinates by means of a radial function $f(\theta, \phi)$. The arguments of this function are the azimuth, $\theta \in [-\pi, \pi]$, and the elevation angles, $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, and the output, r, is the distance between the center of the object and the point on the surface at the corresponding spherical angle pair, i.e., $r = f(\theta, \phi)$. Fig. 3.1 illustrates the representation for an example object.

This representation summarizes the 3D shape exclusively by the external boundary (surface) of the object, considering that the point cloud measurements are merely originated from the surface. Additionally, it implicitly assumes that the latent shape is star-convex[1]. This assumption does not introduce a strict limitation as star-convex shapes present an adequately broad class for object tracking applications.

The main aim of the upcoming sections is to construct a unified state-space model to serve as a basis for the joint estimation of the kinematics and the extent of the object. The corresponding state vector includes both the kinematics and a parametric de-

[1] A set $\mathcal{S}$ is star-convex with respect to the origin if each line segment from the origin to any point in $\mathcal{S}$ is fully contained in $\mathcal{S}$.

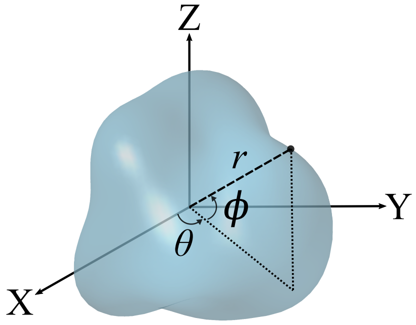


Figure 3.1: Object extent description in spherical coordinates.

scription of the given extent model. This description will be obtained by developing a proper GP model for the radial function. More specifically, we precisely construct the GP model such that it effectively accounts for the inherent spatial correlation within the object surface. Furthermore, we utilize a recursive approximation of GP modeling to avoid associated computational difficulties. The resulting algorithm accomplishes a probabilistic representation of the latent extent in a principled manner. Besides, it is able to maintain the local uncertainty information of the extent which becomes vital for robust tracking and shape learning in scenarios including occlusions and sparse sampling.

The next section first briefly introduces the standard GP regression and then elaborates on its recursive approximation.

3.3 Gaussian Processes

A Gaussian Process (GP) is a stochastic model which specifies a probability distribution in the function space for a function $f(\cdot)$, [42]. We hereby engage a GP to model the radial function expressing the unknown extent of the object. A GP is uniquely defined by the mean $\mu(u)$ and the covariance function $k(u, u')$ defined as[2]

$$\mu(u) = \mathbb{E}[f(u)], \tag{3.1a}$$

$$k(u, u') = \mathbb{E}[(f(u) - \mu(u))(f(u') - \mu(u'))]. \tag{3.1b}$$

[2] Considering the radial distance function, the definition of the GP model is deliberately given for a scalar-valued function.

The corresponding GP model is denoted as

$$f(u) \sim \mathcal{GP}(\mu(u), k(u,u')),$$

where u is the input of the function, which is specified as a scalar for notational simplicity. This model will directly generalize to vector inputs as discussed in Section 3.4.

A GP can also be interpreted as a collection of random variables, any finite number of which have a joint Gaussian distribution that is consistent with the specified mean and covariance functions. The joint distribution of the function evaluations at the inputs, $u_1, ..., u_N$, reads as

$$\begin{bmatrix} f(u_1) \\ \vdots \\ f(u_N) \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, K), \tag{3.2a}$$

where

$$\boldsymbol{\mu} = \begin{bmatrix} \mu(u_1) \\ \vdots \\ \mu(u_N) \end{bmatrix}, K = \begin{bmatrix} k(u_1,u_1) & \ldots & k(u_1,u_N) \\ \vdots & & \vdots \\ k(u_N,u_1) & \ldots & k(u_N,u_N) \end{bmatrix}. \tag{3.2b}$$

3.3.1 Gaussian Process Regression

Prior belief about the unknown function encoded by the GP can be conveniently conditioned on the information provided by observations. For this purpose, a noisy observation m can be described as the true function output perturbed by an independent Gaussian noise e,

$$m = f(u) + e, \;\; e \sim \mathcal{N}(0, R). \tag{3.3}$$

Assume that we seek for the refined distribution of the function values $\mathbf{f} \triangleq [f(u_1^{\mathbf{f}}) \; \ldots \; f(u_{N^{\mathbf{f}}}^{\mathbf{f}})]^\top$ at the inputs $\mathbf{u}^{\mathbf{f}} \triangleq [u_1^{\mathbf{f}} \; \ldots \; u_{N^{\mathbf{f}}}^{\mathbf{f}}]^\top$. Available measurements are denoted by $\mathbf{m} \triangleq [m_1 \; \ldots \; m_N]^\top$ which are originated from the inputs $\mathbf{u} \triangleq [u_1 \; \ldots \; u_N]^\top$. The GP model together with the measurement model (3.3) leads to the following joint distribution,

$$\begin{bmatrix} \mathbf{m} \\ \mathbf{f} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(\mathbf{u}) \\ \boldsymbol{\mu}(\mathbf{u}^{\mathbf{f}}) \end{bmatrix}, \begin{bmatrix} K(\mathbf{u},\mathbf{u}) + I_N R & K(\mathbf{u},\mathbf{u}^{\mathbf{f}}) \\ K(\mathbf{u}^{\mathbf{f}},\mathbf{u}) & K(\mathbf{u}^{\mathbf{f}},\mathbf{u}^{\mathbf{f}}) \end{bmatrix} \right), \tag{3.4a}$$

where

$$\begin{aligned} \boldsymbol{\mu}(\mathbf{u}) &= [\mu(u_1) \ \dots \ \mu(u_N)]^\top, \\ \boldsymbol{\mu}(\mathbf{u}^{\mathbf{f}}) &= [\mu(u_1^{\mathbf{f}}) \ \dots \ \mu(u_{N^{\mathbf{f}}}^{\mathbf{f}})]^\top, \\ K(\mathbf{u},\mathbf{u}^{\mathbf{f}}) &= \begin{bmatrix} k(u_1,u_1^{\mathbf{f}}) & \dots & k(u_1,u_{N^{\mathbf{f}}}^{\mathbf{f}}) \\ \vdots & & \vdots \\ k(u_N,u_1^{\mathbf{f}}) & \dots & k(u_N,u_{N^{\mathbf{f}}}^{\mathbf{f}}) \end{bmatrix}, \end{aligned} \tag{3.4b}$$

I_N indicates an N-by-N identity matrix.

For the sake of brevity, here we will present the case where the mean function of the GP model is zero. However, the method can easily be generalized to an arbitrary mean function and the relevant modifications are presented in Appendix A.1. Under this assumption, the conditional distribution $p(\mathbf{f}|\mathbf{m})$ can be expressed as

$$p(\mathbf{f}|\mathbf{m}) \sim \mathcal{N}(A\mathbf{m}, P), \tag{3.5a}$$

where

$$A = K(\mathbf{u}^{\mathbf{f}},\mathbf{u})K_y^{-1}, \tag{3.5b}$$

$$P = K(\mathbf{u}^{\mathbf{f}},\mathbf{u}^{\mathbf{f}}) - K(\mathbf{u}^{\mathbf{f}},\mathbf{u})K_y^{-1}K(\mathbf{u},\mathbf{u}^{\mathbf{f}}), \tag{3.5c}$$

$$K_y = K(\mathbf{u},\mathbf{u}) + I_N R. \tag{3.5d}$$

3.3.2 Recursive Gaussian Process Regression

The GP regression necessitates to process all available information in a single batch as the complete measurement vector $\mathbf{m}$ and the corresponding covariance matrix K_y appear in (3.5). While this attribute can be interpreted to be the primary strength of GP modeling since it enables to draw conclusions directly from the observations, it also poses some computational problems for certain settings. Specifically, in object tracking, the aim is to compute the posterior density $p(\mathbf{f}|m_{1:k})$ at time k using measurements which are acquired sequentially in time. For this problem, online inference can be achieved by a recursive algorithm which efficiently updates the posterior by considering the newly available measurements. In this respect, the standard GP regression is not applicable due to its increasing needs for computational sources and memory storage with the accumulation of measurements over time. Therefore, we hereby rely on an approximation of the GP, that basically summarizes the original model at a finite set of basis points. The approximation was initially proposed in [48, 49], and then applied to the object tracking problem in [26].

In this approach, the objective is to derive a formulation of the posterior distribution $p(\mathbf{f}|m_{1:N})$ that enables recursive implementation. To this end, the posterior is first expanded as the collection of the following terms by applying the Bayes' law iteratively.

$$\begin{aligned} p(\mathbf{f}|m_{1:N}) &\propto p(m_N|\mathbf{f}, m_{1:N-1})p(\mathbf{f}|m_{1:N-1}) & \text{(3.6a)} \\ &\propto p(\mathbf{f})p(m_1|\mathbf{f}) \prod_{k=2}^{N} p(m_k|\mathbf{f}, m_{1:k-1}) & \text{(3.6b)} \end{aligned}$$

Assumption: $\mathbf{f}$ provides the sufficient statistics for m_k.

Under this assumption, m_k conditioned on $\mathbf{f}$ becomes independent from all previous measurements, $m_{1:k-1}$, i.e.,

$$p(m_k|\mathbf{f}, m_{1:k-1}) \approx p(m_k|\mathbf{f}). \tag{3.7}$$

Notice that the assumption becomes exact if the inputs of m_k form a subset of the inputs of $\mathbf{f}$. Moreover, it can be claimed to be a reasonable approximation when the distance between the inputs of m_k and the inputs of $\mathbf{f}$ is sufficiently small compared to the characteristic length-scale of the covariance function. In this chapter, we want to model the unknown radial function whose input is the spherical angle pair. As the set of the possible input values has a well-defined boundary, it is possible to sufficiently sample this set by a finite number of basis points which will be located equidistantly.

The above assumption leads to a setting where we essentially treat $\mathbf{f}$ to be the latent variable and the measurements provide noisy observations of it. Accordingly, once the measurement likelihood and the initial prior densities are defined, it is possible to apply recursive Bayesian inference for $\mathbf{f}$. With this in mind, we simply refer to the underlying GP model to offer these densities in a principled way. At first, the joint distribution of the measurement m_k and $\mathbf{f}$ is revealed by the definition of the GP as

$$\begin{bmatrix} m_k \\ \mathbf{f} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(u_k, u_k) + R & K(u_k, \mathbf{u}^{\mathbf{f}}) \\ K(\mathbf{u}^{\mathbf{f}}, u_k) & K(\mathbf{u}^{\mathbf{f}}, \mathbf{u}^{\mathbf{f}}) \end{bmatrix}\right). \tag{3.8}$$

Then, the joint distribution immediately offers the following likelihood and prior densities,

$$p(m_k|\mathbf{f}) = \mathcal{N}(m_k; H_k^{\mathbf{f}}\mathbf{f}, R_k^{\mathbf{f}}), \tag{3.9a}$$

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{0}, P_0^{\mathbf{f}}), \tag{3.9b}$$

where

$$H_k^{\mathbf{f}} = H^{\mathbf{f}}(u_k) = K(u_k, \mathbf{u}^{\mathbf{f}})[K(\mathbf{u}^{\mathbf{f}}, \mathbf{u}^{\mathbf{f}})]^{-1}, \tag{3.9c}$$

$$R_k^{\mathbf{f}} = R^{\mathbf{f}}(u_k) = k(u_k, u_k) + R - K(u_k, \mathbf{u}^{\mathbf{f}})[K(\mathbf{u}^{\mathbf{f}}, \mathbf{u}^{\mathbf{f}})]^{-1}K(\mathbf{u}^{\mathbf{f}}, u_k), \tag{3.9d}$$

$$P_0^{\mathbf{f}} = K(\mathbf{u}^{\mathbf{f}}, \mathbf{u}^{\mathbf{f}}). \tag{3.9e}$$

The structure of (3.9) allows us to construct the following state-space model to which a standard Kalman filter can be applied for recursive inference, [26].

$$\mathbf{f}_{k+1} = \mathbf{f}_k, \tag{3.10a}$$

$$m_k = H_k^{\mathbf{f}}\,\mathbf{f}_k + e_k^{\mathbf{f}}, \quad e_k^{\mathbf{f}} \sim \mathcal{N}(0, R_k^{\mathbf{f}}), \tag{3.10b}$$

$$\mathbf{f}_0 \sim \mathcal{N}(\mathbf{0}, P_0^{\mathbf{f}}), \tag{3.10c}$$

where $\mathbf{f}_k$ is basically defined as the latent function values evaluated at the predetermined inputs, i.e., $\mathbf{f}_k \triangleq \mathbf{f}$.

The benefits of having such a state-space model for the object extent are twofold: first, we can easily engage a dynamical model in (3.10a) to express the time evolution of the extent; second, it can simply be augmented by another state-space model to obtain a unified representation. We will basically benefit from these advantages while developing the unified state-space model in Section 3.5.

3.4 GP Modeling of Object Extent

In this section, the radial function which expresses the object extent is to be modeled via a GP. By doing so, we will be able to facilitate effective shape learning in the probabilistic framework by using incomplete and noisy point measurements.

GP lends itself conveniently to extent modeling since it is naturally able to describe the spatial correlation between different sections the object surface. In addition, it maintains local uncertainty information associated with the object surface which is critical for accurate gating and association of the measurements leading to robust tracking performance.

A GP model is uniquely defined by its mean and covariance functions, hence the main focus of this discussion is to rigorously construct these functions regarding the characteristics of the extent representation. Note that we hereby put forward a generic approach to be able to apply to arbitrarily shaped objects; however, prior knowledge about the object shape can also be systematically incorporated by adjusting these functions accordingly, e.g., see [56].

As discussed earlier, the output of the radial function is the distance r, and the input is the pair of azimuth and elevation angles (θ, ϕ), i.e., $r = f(\theta, \phi)$. For notational simplicity, the pair (θ, ϕ) is assigned to $\boldsymbol{\gamma}$, i.e., $\boldsymbol{\gamma} \triangleq (\theta, \phi)$ and $r = f(\boldsymbol{\gamma})$. Therefore, we denote the mean and covariance functions as $\mu(\boldsymbol{\gamma})$ and $k(\boldsymbol{\gamma}, \boldsymbol{\gamma}')$, respectively, and $f(\boldsymbol{\gamma}) \sim \mathcal{GP}(\mu(\boldsymbol{\gamma}), k(\boldsymbol{\gamma}, \boldsymbol{\gamma}'))$ indicates the GP model.

3.4.1 Mean Function

The mean function of the GP model is assumed to be an unknown constant having a normal distribution, i.e.,

$$\mu(\boldsymbol{\gamma}) = r, \quad \text{where } r \sim \mathcal{N}(\mu_r, \sigma_r^2). \tag{3.11}$$

Additionally, we can obviously express the original GP model as in

$$f(\boldsymbol{\gamma}) = \bar{f}(\boldsymbol{\gamma}) + \mu(\boldsymbol{\gamma}), \text{ where } \bar{f}(\boldsymbol{\gamma}) \sim \mathcal{GP}(0, k(\boldsymbol{\gamma}, \boldsymbol{\gamma}')). \tag{3.12}$$

Consequently, by using the prior distribution of $\mu(\boldsymbol{\gamma})$ in (3.11), we can obtain an equivalent representation of (3.12) as follows, [42, Ch. 2.7], [57].

$$f(\boldsymbol{\gamma}) \sim \mathcal{GP}(\mu_r, k_{total}(\boldsymbol{\gamma}, \boldsymbol{\gamma}')), \text{ where } \quad k_{total}(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = k(\boldsymbol{\gamma}, \boldsymbol{\gamma}') + \sigma_r^2. \tag{3.13}$$

3.4.2 Covariance Function

Selection of the covariance function for a GP is of great importance since it basically determines the characteristics of the functions to be learned. In this application, it is required to conform to the fundamentals of 3D object geometry as it encodes the spatial correlation between points on the extent.

The design of the covariance function is initiated from the exponentiated quadratic function, which is accepted to be the de facto choice in various fields, [42],

$$k(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \sigma_f^2 e^{-\frac{d^2(\boldsymbol{\gamma}, \boldsymbol{\gamma}')}{2l^2}}, \tag{3.14}$$

where σ_f^2 stands for the prior variance, l is the length-scale and $d(\boldsymbol{\gamma}, \boldsymbol{\gamma}')$ calculates the relative distance between two inputs. The prior variance specifies the typical amount of variation observed among the functions sampled from a GP. Accordingly, increasing prior variance in the proposed extent model would imply that there is relatively less prior information about the size of the objects to be tracked. Additionally, the length scale determines the 'smoothness' of the functions modeled by a GP. In this context, a shorter length-scale would suggest that the corresponding extent is potentially sharper (exhibiting high-frequency oscillations) while increasing the length-scale would lead to a smoother extent representation.

The unconventional aspect of the employed covariance function is the formulation of the distance, $d(\boldsymbol{\gamma}, \boldsymbol{\gamma}')$. It is specified to imply higher correlation for closer regions compared to those which are rather separated. An immediate option for the distance definition could be the Euclidean distance, i.e., $d(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \|\boldsymbol{\gamma} - \boldsymbol{\gamma}'\|$, as used in [26]. However, being inconsistent with the basics of the spherical geometry, it leads to erroneous correlation patterns for the extent defined in the spherical coordinates. As a simple example, consider $\boldsymbol{\gamma} = \big(0, \frac{\pi}{2}\big)$ and $\boldsymbol{\gamma}' = \big(\pi, \frac{\pi}{2}\big)$ both pointing to the upper pole of a sphere. For these inputs, the Euclidean distance is computed as π which is also equal to the distance for any two spherical angles pointing to opposite directions, e.g., the upper and the lower poles, i.e., $\boldsymbol{\gamma} = \big(0, \frac{\pi}{2}\big)$ and $\boldsymbol{\gamma}' = \big(0, -\frac{\pi}{2}\big)$. Such problems encountered in the angular estimation applications are commonly addressed by suitable angular distance measures in the literature (see, for example, [58] and the references therein). In this chapter, we suggest an alternative distance definition that naturally induces a proper correlation structure. Consider two unit-length vectors expressed by the spherical angles pairs, $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}'$. We set $d(\boldsymbol{\gamma}, \boldsymbol{\gamma}')$ to be the angle between these two vectors. From another viewpoint, this definition is the angle corresponding to the shortest arc on a sphere that connects the two points described by $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}'$. The analytical expression for the definition can be written as

$$\begin{aligned} d(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = \arccos\big(&\cos(\phi)\cos(\phi')\cos(\theta)\cos(\theta') \\ &+ \cos(\phi)\cos(\phi')\sin(\theta)\sin(\theta') + \sin(\phi)\sin(\phi')\big), \end{aligned} \tag{3.15}$$

where $\boldsymbol{\gamma} = (\theta, \phi)$ and $\boldsymbol{\gamma}' = (\theta', \phi')$. Notice that with this formulation, the distance takes values within the interval $[0, \pi]$, and any coincident angle pair is mapped to 0 while opposite directions compute π.

Finally, the total covariance function in (3.13) is attained as

$$\begin{aligned} k_{total}(\boldsymbol{\gamma}, \boldsymbol{\gamma}') &= k(\boldsymbol{\gamma}, \boldsymbol{\gamma}') + \sigma_r^2, \\ &= \sigma_f^2 e^{-\frac{d^2(\boldsymbol{\gamma}, \boldsymbol{\gamma}')}{2l^2}} + \sigma_r^2. \end{aligned} \tag{3.16}$$

The formulation of the covariance function can easily be adapted regarding the characteristics of a specific application. For instance, let us consider a setting where an autonomous robotic manipulator is assigned to grasp the surrounding objects with unknown shapes. If there is prior information available regarding the environment, e.g., the objects of interest are symmetric around their azimuth axis, then one can account for this specification by adjusting the utilized distance function accordingly.

3.5 State-Space Model

In this section, we will develop a state-space model to be used in object tracking. This model is based on the state vector involving both the kinematics and the extent representation of the object. In this setting, joint estimation of this aggregated state variables will be accomplished by a single inference algorithm. In other words, the idea of leveraging the latent shape information for object tracking is basically realized by this formulation.

The state vector is defined as $\mathbf{x}_k \triangleq \left[\bar{\mathbf{x}}_k^\top \; \mathbf{f}_k^\top\right]^\top$ where $\bar{\mathbf{x}}_k$ consists of the translational, $\mathbf{x}_k^t$, and the rotational, $\mathbf{x}_k^r$, kinematic variables, i.e., $\bar{\mathbf{x}}_k^\top \triangleq \left[\mathbf{x}_k^{t\top} \; \mathbf{x}_k^{r\top}\right]^\top$, and $\mathbf{f}_k$ indicates the extent representation.

The definition of the state vector makes use of two distinct coordinate frames as shown in Fig. 3.2. The first one is the global coordinate frame which is fixed to the sensor; the second one is the local coordinate frame which is attached to the object to be tracked. As the local frame performs exactly the same motion with the object, it allows to describe the extent in a consistent manner. Accordingly, the extent information is maintained in the local coordinate frame while the object motion is estimated in the global coordinate frame.

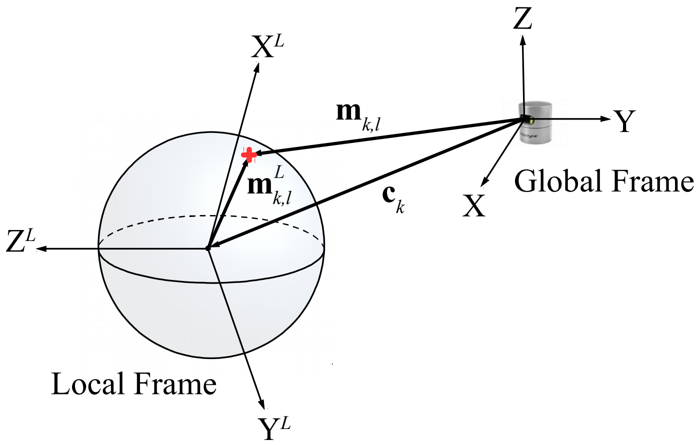


Figure 3.2: Illustration of the coordinate frames and the vectors regarded in the measurement model.

An overview of the state-space model is given by the following set of equations,

$$\mathbf{x}_{k+1} = F_k \mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q_k), \tag{3.17a}$$

$$\mathbf{0} = \mathbf{h}(\mathbf{x}_k, \mathbf{m}_{k,l}) + \mathbf{e}_{k,l}, \quad \mathbf{e}_{k,l} \sim \mathcal{N}(\mathbf{0}, R_{k,l}), \tag{3.17b}$$

$$\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, P_0), \tag{3.17c}$$

where k is the time index; $\mathbf{w}_k$ and $\mathbf{e}_{k,l}$ indicate the zero-mean process and the measurement noise, respectively. These are assumed to be Gaussian with covariance matrices defined as $\mathrm{cov}[\mathbf{w}_k] = Q_k$ and $\mathrm{cov}[\mathbf{e}_{k,l}] = R_{k,l}$. The following subsections will introduce the details of these equations starting from the derivation of the process model.

3.5.1 Process Model

The overview of the process model, which describes the evolution of the states over time, is given by the linear Gaussian model in (3.17a). Recall that the state vector is formed by concatenating the kinematics and the extent representation, i.e., $\mathbf{x}_k \triangleq \left[\bar{\mathbf{x}}_k^\top \; \mathbf{f}_k^\top\right]^\top$. It is assumed that the dynamics of these two state components do not interact with each other. Therefore, the process model includes two independent subsystems and can explicitly be written as

$$F_k = \begin{bmatrix} \bar{F}_k & 0 \\ 0 & F^{\mathbf{f}} \end{bmatrix}, \; Q_k = \begin{bmatrix} \bar{Q}_k & 0 \\ 0 & Q_k^{\mathbf{f}} \end{bmatrix}. \tag{3.18}$$

Accordingly, the prior distribution of the state given in (3.17c) can be specified by

$$\boldsymbol{\mu}_0 = \begin{bmatrix} \bar{\boldsymbol{\mu}}_0 \\ \boldsymbol{\mu}_0^{\mathbf{f}} \end{bmatrix}, \; P_0 = \begin{bmatrix} \bar{P}_0 & 0 \\ 0 & P_0^{\mathbf{f}} \end{bmatrix}. \tag{3.19}$$

Our formulation does not put any restriction on the selection of the process model; hence one can freely design the model for both the kinematics and the extent accounting for the characteristics of a specific application.

We hereby use the following dynamical model for the object extent:

$$\mathbf{f}_{k+1} = \mathbf{f}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q_k^{\mathbf{f}}), \tag{3.20a}$$

where

$$Q_k^{\mathbf{f}} = \left(\frac{1}{\lambda} - 1\right) P_{k|k}^{\mathbf{f}}, \tag{3.20b}$$

which implies that $F^{\mathbf{f}}$ is set to be an identity matrix. $P_{k|k}^{\mathbf{f}}$ denotes the covariance of the estimated extent state. Notice that the prediction density computed using this model has the same mean with that of the estimated density, while the prediction covariance is scaled up as $P_{k+1|k}^{\mathbf{f}} = \frac{1}{\lambda} P_{k|k}^{\mathbf{f}}$ for $\lambda < 1$. The model provides a maximum entropy distribution for the extent prediction, as shown in [59]. This approach reflects that the transition density is unknown, while the Kullback-Leibler divergence to the prediction density is upper bounded.

With this model, we can basically account for the possible changes in the object extent. Therefore, it potentially facilitates the tracking of nonrigid objects. In addition, it enables recovery from erroneously integrated shape information, which might occur due to temporal errors in the pose estimates, especially during the initial phases of the tracking.

In this work, we assume that the rotational and the translational motion can be modeled independently. Therefore, $\bar{F}_k$ and $\bar{Q}_k$ can be written as

$$\bar{F}_k = \begin{bmatrix} F^t & 0 \\ 0 & F^r_k \end{bmatrix}, \; \bar{Q}_k = \begin{bmatrix} Q^t & 0 \\ 0 & Q^r_k \end{bmatrix}. \tag{3.21}$$

3.5.1.1 Translational Motion Model

We employ the well-known almost constant velocity model for the translational motion as given in (3.22). The translational kinematic state is defined as $\mathbf{x}^t_k \triangleq \begin{bmatrix} \mathbf{c}^\top_k & \mathbf{v}^\top_k \end{bmatrix}^\top$ where $\mathbf{c}_k \in \mathbb{R}^3$ is position of the object center, and $\mathbf{v}_k \in \mathbb{R}^3$ stands for the velocity of the center.

$$\mathbf{x}^t_{k+1} = F^t \mathbf{x}^t_k + \mathbf{w}^t_k, \quad \mathbf{w}^t_k \sim \mathcal{N}(\mathbf{0}, Q^t), \tag{3.22a}$$

$$\mathbf{x}^t_0 \sim \mathcal{N}(\boldsymbol{\mu}^t_0, P^t_0), \tag{3.22b}$$

where

$$F^t = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes I_3, \; Q^t = \begin{bmatrix} \frac{T}{3}^3 & \frac{T}{2}^2 \\ \frac{T}{2}^2 & T \end{bmatrix} \otimes (\sigma^2_c I_3). \tag{3.22c}$$

σ^2_c is the process noise variance for the center, $\otimes$ is the Kronecker product, and T is sampling time.

3.5.1.2 Rotational Motion Model

The representation of the orientation of the object is of paramount importance as it directly affects the performance of tracking and shape estimation. In the literature, there are many alternative representations for the orientation of 3D objects. The main challenges associated with these are the included singularities and the inherent constraints, [60]. For example, expressing the orientation using three variables, Euler angles offer a minimal representation. However, this approach is impeded by the singularities, and it can not provide a global orientation description. On the other hand, the four-component unit quaternion presents the lowest dimensional representation that avoids any possible singularity. Nevertheless, it introduces a nonlinear equality constraint such that the norm of the quaternion has to be always exactly one. Hence, it requires special treatment during the estimation process.

Considering the aforementioned difficulties, we rely on an alternative parametrization of the orientation. This representation stores a reference orientation as a unit quaternion, and the orientation deviation from this reference is expressed by a three-component error vector. By this approach, a standard extended Kalman filter (EKF) can easily be employed to estimate the error vector, and the reference orientation is periodically updated by the estimated deviation. The resulting algorithm is referred to as multiplicative EKF (MEKF), [61]. The virtue of the method is twofold: first, it can represent the orientation globally as it essentially exploits a unit quaternion description; second, the norm constraint of the unit quaternion is naturally satisfied as the quaternion gets updated appropriately by the error vector.

In the rest of this subsection, we will rely on this representation to derive a constant velocity model for the object orientation. The notation and the basics of the quaternion algebra is adopted from [62], which closely follows [63].

The unit quaternion $\mathbf{q} \in \mathbb{R}^4$ is defined as

$$\mathbf{q} \triangleq \begin{bmatrix} q_1 \; q_2 \; q_3 \; q_4 \end{bmatrix}^\top = \begin{bmatrix} \bar{\mathbf{q}}^\top \; q_4 \end{bmatrix}^\top , \tag{3.23}$$

where $\bar{\mathbf{q}} \triangleq [q_1 \; q_2 \; q_3]^\top$ and $|\mathbf{q}|^2 = |\bar{\mathbf{q}}|^2 + q_4^2 = 1$.

Then, the rotation matrix $R_G^L(\mathbf{q})$ that expresses the orientation of the local frame with respect to the global frame is given by

$$R_L^G(\mathbf{q}) = (q_4^2 - \bar{\mathbf{q}}^\top \bar{\mathbf{q}}) I_3 + 2\bar{\mathbf{q}}\bar{\mathbf{q}}^\top - 2q_4 [\bar{\mathbf{q}}\times] , \tag{3.24}$$

where $[\bar{\mathbf{q}}\times]$ is the cross product matrix

$$[\bar{\mathbf{q}}\times] = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} . \tag{3.25}$$

The quaternion product is defined as

$$\mathbf{p} \odot \mathbf{q} = \begin{bmatrix} p_4\bar{\mathbf{q}} + q_4\bar{\mathbf{p}} - \bar{\mathbf{p}} \times \bar{\mathbf{q}} \\ p_4 q_4 - \bar{\mathbf{p}}^\top \bar{\mathbf{q}} \end{bmatrix}, \tag{3.26}$$

which leads to the following useful property

$$R(\mathbf{p})R(\mathbf{q}) = R(\mathbf{p} \odot \mathbf{q}) , \tag{3.27}$$

where $R(\mathbf{p})$ and $R(\mathbf{q})$ denote the rotation matrices corresponding to the unit quaternions $\mathbf{p}$ and $\mathbf{q}$, respectively. This equation reveals that successive rotation operations can be expressed in terms of the quaternion product. Based on this fact, we can describe the orientation of the object as follows

$$\mathbf{q} = \delta\mathbf{q}(\mathbf{a}) \odot \mathbf{q}_{\text{ref}} , \tag{3.28}$$

where $\mathbf{q}_{\text{ref}}$ indicates a reference orientation, and $\delta\mathbf{q}(\cdot)$ corresponds to the deviation from the reference. $\delta\mathbf{q}(\mathbf{a})$ is defined via the Rodrigues parametrization, i.e.,

$$\delta\mathbf{q}(\mathbf{a}) = \frac{1}{\sqrt{4 + |\mathbf{a}|^2}} \begin{bmatrix} \mathbf{a} \\ 2 \end{bmatrix}, \tag{3.29}$$

where $\mathbf{a} \in \mathbb{R}^3$.

The central idea of this approach is to treat the deviation vector, $\mathbf{a}$, as the latent variable and to estimate it by means of nonlinear Bayesian filtering. To this end, a process model for the orientation deviation is needed. In [63], it has been shown that the continuous time dynamics of the deviation vector can be written as

$$\dot{\mathbf{a}} = \left(I_3 + \frac{1}{4}\mathbf{a}\mathbf{a}^\top + \frac{1}{2}[\mathbf{a}\times] \right) \boldsymbol{\omega} , \tag{3.30}$$

where $\boldsymbol{\omega} \triangleq [\omega_x \; \omega_y \; \omega_z]^\top$ represents the angular rate of the local frame with respect to the global frame. We assume that $\mathbf{a}$ takes small values and approximate the expression as

$$\dot{\mathbf{a}} \approx \left(I_3 + \frac{1}{2}[\mathbf{a}\times] \right) \boldsymbol{\omega} . \tag{3.31}$$

Based on (3.31), a constant velocity model can easily be specified in continuous time,

$$\dot{\begin{bmatrix} \mathbf{a} \\ \boldsymbol{\omega} \end{bmatrix}} = \begin{bmatrix} \left(I_3 + \frac{1}{2}[\mathbf{a}\times]\right)\boldsymbol{\omega} \\ 0_{3\times1} \end{bmatrix} + \begin{bmatrix} 0_3 \\ I_3 \end{bmatrix} \boldsymbol{\alpha} , \tag{3.32}$$

where $\boldsymbol{\alpha}$ denotes the rotational acceleration vector, and it is modeled as zero-mean white Gaussian noise with covariance $\mathrm{cov}[\boldsymbol{\alpha}(t), \boldsymbol{\alpha}(t')] = \delta(t - t')\Sigma_\alpha$, where $\Sigma_\alpha = \sigma_\alpha^2 I_3$.

We need to discretize this system to be able to plug it in (3.21). Notice that (3.32) is a nonlinear dynamic model. Therefore, we will first linearize the equation around some point of interest and then discretize the resulting linearized model. Let us first rewrite (3.32) as

$$\dot{\mathbf{x}}^r = f(\mathbf{x}^r) + B\boldsymbol{\alpha}\ ,$$

where $\mathbf{x}^r \triangleq \left[\mathbf{a}^\top\ \boldsymbol{\omega}^\top\right]^\top$, and $f(\cdot)$ and B are determined accordingly. To linearize the model, we will substitute $f(\cdot)$ by its first order Taylor series approximation.

$$f(\mathbf{x}^r) \approx f(\hat{\mathbf{x}}^r) + A_k^r(\mathbf{x}^r - \hat{\mathbf{x}}^r), \tag{3.33a}$$

where

$$A_k^r = \frac{d}{d\mathbf{x}^r} f(\mathbf{x}^r)|_{\mathbf{x}^r = \hat{\mathbf{x}}_{k|k}^r}\ . \tag{3.33b}$$

The linearization is performed around the best available point estimate which is the mean of the previous posterior, i.e., $\hat{\mathbf{x}}_{k|k}^r = \left[\hat{\mathbf{a}}_{k|k}^\top\ \hat{\boldsymbol{\omega}}_{k|k}^\top\right]^\top$. Note that after each measurement update of the filter, the reference orientation is updated by the estimated orientation deviation using the quaternion product, and then the deviation vector is reset to zero (see Appendix A.3 for the details). Therefore, $\hat{\mathbf{a}}_{k|k}$ is equal to the zero vector. Consequently, the Taylor series approximation in (3.33) reduces to

$$f(\mathbf{x}^r) = A_k^r \mathbf{x}^r, \tag{3.34a}$$

where

$$A_k^r = \begin{bmatrix} \frac{1}{2}[-\hat{\boldsymbol{\omega}}_{k|k}\times] & I_3 \\ 0_3 & 0_3 \end{bmatrix}. \tag{3.34b}$$

The resulting linearized system is given by

$$\dot{\mathbf{x}}^r = A_k^r\, \mathbf{x}^r + B\boldsymbol{\alpha}. \tag{3.35}$$

Thereafter, we discretize this equation and end up with the following linear Gaussian model which expresses the dynamics of the rotational subsystem.

$$\mathbf{x}_{k+1}^r = F_k^r \mathbf{x}_k^r + \mathbf{w}_k^r, \quad \mathbf{w}_k^r \sim \mathcal{N}(\mathbf{0}, Q_k^r), \tag{3.36a}$$

where

$$F^r_k = \exp(A^r_k T)\ , \tag{3.36b}$$

$$Q^r_k = G_k \Sigma_\alpha G_k^\top\ , \tag{3.36c}$$

$$G_k = \left(\int_0^T \exp(A^r_k \tau) d\tau \right) B\ . \tag{3.36d}$$

$\mathbf{x}^r_k \triangleq \left[\mathbf{a}_k^\top \ \boldsymbol{\omega}_k^\top \right]^\top$ is the rotational kinematic state, F^r_k is the system matrix, Q^r_k is the process noise covariance matrix, and T is the sampling time. F^r_k and Q^r_k are explicitly indicated to be time-varying as they are recalculated considering the new linearization point at each iteration of the filter. For the complete details of the matrices in (3.36), see Appendix A.4.

The derivation of the given discrete-time state-space model follows a standard approach, see, for example, [64, Ch. 12.2]. It is well known that the fidelity of the resulting model depends on the accuracy of the rotational state estimate as F^r_k and Q^r_k are approximately computed regarding the local linearization in (3.33).

3.5.2 Measurement Model

In this subsection, the measurement model expressing the relation between the measurements and the state variables is derived to complete the state-space model. In general, we assume that there are multiple point measurements returned from an object at time k, which can be represented by the set $\{\mathbf{m}_{k,l}\}_{l=1}^{n_k}$. A single measurement can be expressed as

$$\mathbf{m}_{k,l} = \mathbf{c}_k + \mathbf{p}_{k,l}\ f((\theta,\phi)_{k,l}) + \bar{\mathbf{e}}_{k,l},\ \bar{\mathbf{e}}_{k,l} \sim \mathcal{N}(\mathbf{0}, \bar{R}). \tag{3.37}$$

$\mathbf{c}_k$ is the center of the object at time k; $(\theta,\phi)_{k,l}$ is the spherical angle pair indicating the measurement source on the object surface that originates $\mathbf{m}_{k,l}$; $\mathbf{p}_{k,l}$ is the unit-length vector that starts from the object center and points towards the measurement source; $f(\cdot)$ is the radial function; and $\bar{\mathbf{e}}_{k,l}$ stands for the zero-mean Gaussian measurement noise with covariance $\bar{R}$.

Notice that for the measurement $\mathbf{m}_{k,l}$ in (3.37), the underlying measurement source is unknown, and hence the corresponding $\mathbf{p}_{k,l}$ and $(\theta,\phi)_{k,l}$ are not available. As an approximate approach, we hereby express these variables by utilizing the kinematic variables and the measurement.

Firstly, $\mathbf{p}_{k,l}(\mathbf{c}_k, \mathbf{m}_{k,l})$ is defined to be the unit-length vector starting from the object center and pointing to the measurement as

$$\mathbf{p}_{k,l}(\mathbf{c}_k, \mathbf{m}_{k,l}) = \frac{\mathbf{m}_{k,l} - \mathbf{c}_k}{\|\mathbf{m}_{k,l} - \mathbf{c}_k\|}. \tag{3.38}$$

Then, for $(\theta, \phi)_{k,l}$ we need an intermediate representation of $\mathbf{m}_{k,l}$ by resolving it in the local coordinate frame. It is simply obtained by the successive transformations of translation and rotation as

$$\mathbf{m}^L_{k,l}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l}) = \underbrace{R^L_G(\mathbf{q}_k)}_{Rotation} \underbrace{(\mathbf{m}_{k,l} - \mathbf{c}_k)}_{Translation}. \tag{3.39}$$

$R^L_G(\mathbf{q})$ is the rotation matrix from the global to the local frame.

The relation between $\mathbf{m}_{k,l}$ and $\mathbf{m}^L_{k,l}$ is illustrated in Fig. 3.2. $\mathbf{m}^L_{k,l}$ can be interpreted as a phantom measurement in the local frame, and it will only be exploited to find out the spherical angle pair, $(\theta_{k,l}, \phi_{k,l})$, associated to $\mathbf{m}_{k,l}$. The figure depicts the object by a spherical shape to provide a straightforward description although the procedure applies to any arbitrary object. Then, $(\theta_{k,l}, \phi_{k,l})$ is easily computed by converting $\mathbf{m}^L_{k,l}$ into the spherical coordinates by

$$\theta_{k,l} = \arctan\left(y^L/x^L\right), \tag{3.40a}$$

$$\phi_{k,l} = \arctan\left(z^L/\sqrt{(x^L)^2 + (y^L)^2}\right), \tag{3.40b}$$

where $\mathbf{m}^L_{k,l} \triangleq (x^L, y^L, z^L)$.

Subsequently, the measurement equation in (3.37) is rewritten by

$$\mathbf{m}_{k,l} = \mathbf{c}_k + \mathbf{p}_{k,l}(\mathbf{c}_k, \mathbf{m}_{k,l}) f\left(\boldsymbol{\gamma}_{k,l}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l})\right) + \bar{\mathbf{e}}_{k,l}\ , \tag{3.41}$$

where the spherical angle pair is indicated by $\boldsymbol{\gamma}_{k,l} \triangleq (\theta_{k,l}, \phi_{k,l})$ for brevity. Note that $\boldsymbol{\gamma}_{k,l}$ is a function of $\mathbf{c}_k$, $\mathbf{q}_k$ and $\mathbf{m}_{k,l}$ as implied by (3.39) and (3.40).

Finally, the GP representation for the radial function given in (3.10) is substituted, and we end up with the following measurement model.

$$\begin{aligned}
\mathbf{m}_{k,l} &= \mathbf{c}_k + \mathbf{p}_{k,l}\left[H^{\mathbf{f}}\left(\boldsymbol{\gamma}_{k,l}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l})\right)\mathbf{f}_k + e^{\mathbf{f}}_{k,l}\right] + \bar{\mathbf{e}}_{k,l} \\
&= \underbrace{\mathbf{c}_k + \tilde{H}\left(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l}\right)\mathbf{f}_k}_{=\tilde{\mathbf{h}}(\mathbf{x}_k, \mathbf{m}_{k,l})} + \underbrace{\mathbf{p}_{k,l}\, e^{\mathbf{f}}_{k,l} + \bar{\mathbf{e}}_{k,l}}_{=\mathbf{e}_{k,l}} \\
&= \tilde{\mathbf{h}}(\mathbf{x}_k, \mathbf{m}_{k,l}) + \mathbf{e}_{k,l}, \quad \mathbf{e}_{k,l} \sim \mathcal{N}(\mathbf{0}, R_{k,l}), \qquad (3.42a)
\end{aligned}$$

where

$$\begin{aligned}
\tilde{H}\left(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l}\right) &= \mathbf{p}_{k,l}\, H^{\mathbf{f}}\left(\boldsymbol{\gamma}_{k,l}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l})\right), \qquad (3.42b) \\
R_{k,l} &= \mathbf{p}_{k,l}\, R^{\mathbf{f}}_{k,l}\, \mathbf{p}^{\top}_{k,l} + \bar{R}\,, \qquad (3.42c) \\
\mathbf{p}_{k,l} &= \mathbf{p}_{k,l}(\mathbf{c}_k, \mathbf{m}_{k,l})\,, \qquad (3.42d) \\
R^{\mathbf{f}}_{k,l} &= R^{\mathbf{f}}\left(\boldsymbol{\gamma}_{k,l}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l})\right). \qquad (3.42e)
\end{aligned}$$

Notice that the additive noise term $\mathbf{e}_{k,l}$ in (3.42a) actually depends on the unknown state vector, and it is not necessarily Gaussian. However, we deliberately ignore this dependence and assume a Gaussian density to form an approximate model that is appropriate for the employed inference scheme.

(3.42a) implies an implicit measurement model as $\mathbf{m}_{k,l}$ can not be explicitly written as a function of the state vector and the measurement noise. We collect all the terms on one side of the equation as

$$\begin{aligned}
\mathbf{0} &= \underbrace{-\mathbf{m}_{k,l} + \tilde{\mathbf{h}}(\mathbf{x}_k, \mathbf{m}_{k,l})}_{=\mathbf{h}(\mathbf{x}_k, \mathbf{m}_{k,l})} + \mathbf{e}_{k,l} \\
&= \mathbf{h}(\mathbf{x}_k, \mathbf{m}_{k,l}) + \mathbf{e}_{k,l}, \quad \mathbf{e}_{k,l} \sim \mathcal{N}(\mathbf{0}, R_{k,l}). \qquad (3.43)
\end{aligned}$$

The zero vector in (3.43) can be interpreted as a pseudo-measurement, which is a nonlinear function of the state and the measurement, and it is corrupted by some additive Gaussian noise. Implicit measurement models have been commonly utilized for different applications, which perform inference by various means of Kalman filtering (see, for example, [13], [65]).

3.6 Inference

Having developed the state-space model, the last step is to design an effective inference method to realize object tracking using point cloud measurements. While there are various standard techniques to recursively compute the posterior distribution of the state vector, we employ an extended Kalman filter (EKF) due to the nonlinearities in the measurement model.

To be able to process multiple measurements $\{\mathbf{m}_{k,l}\}_{l=1}^{n_k}$ in a single recursion at time k, we first need to slightly modify the state-space model. To this end, the following measurement vector is created by concatenating the measurements together,

$$\mathbf{m}_k = \left[\mathbf{m}_{k,1}^\top, \; \ldots, \; \mathbf{m}_{k,n_k}^\top\right]^\top. \tag{3.44}$$

Then, the corresponding measurement equation can be simply written as

$$\begin{aligned} \mathbf{0} &= \mathbf{h}(\mathbf{x}_k, \mathbf{m}_k) + \mathbf{e}_k, \quad \mathbf{e}_k \sim \mathcal{N}(\mathbf{0}, R_k), && \text{(3.45a)} \\ \mathbf{h}(\mathbf{x}_k, \mathbf{m}_k) &= \left[\mathbf{h}(\mathbf{x}_k, \mathbf{m}_{k,1})^\top, \; \ldots, \; \mathbf{h}(\mathbf{x}_k, \mathbf{m}_{k,n_k})^\top\right]^\top, && \text{(3.45b)} \\ R_k &= \mathrm{diag}\left[R_{k,1}, \; \ldots, \; R_{k,n_k}\right]. && \text{(3.45c)} \end{aligned}$$

Notice that R_k is formed as a block diagonal matrix by considering that the noise coupled to the individual measurements are mutually independent.

Consequently, the state-space model considering the complete set of measurements reads as

$$\begin{aligned} \mathbf{x}_{k+1} &= F_k \mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q_k), && \text{(3.46a)} \\ \mathbf{0} &= \mathbf{h}(\mathbf{x}_k, \mathbf{m}_k) + \mathbf{e}_k, \quad \mathbf{e}_k \sim \mathcal{N}(\mathbf{0}, R_k), && \text{(3.46b)} \\ \mathbf{x}_0 &\sim \mathcal{N}(\boldsymbol{\mu}_0, P_0). && \text{(3.46c)} \end{aligned}$$

The EKF regards the above representation to recursively compute the estimate of the state vector, $\hat{\mathbf{x}}_{k|k}$. Note that the gradient of the measurement function $\frac{\partial \mathbf{h}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k}$ can be derived analytically which is to be utilized in the measurement update phase of each recursion.

For very large point clouds, one may want to optimize the computational characteristics of the filter, which is mainly determined by the inversion operation of the innovation covariance matrix. In this regard, a sequential update of the measurements can be preferred instead of a batch update. The update can also be performed in the information form of the EKF, [66, Ch. 3.5].

3.7 3D Extent Tracking Using Projections

In the first part of this chapter, we developed a tracking algorithm which is essentially based on the radial function representation, $f(\theta, \phi)$, of the underlying 3D object shape. This function is further approximated via some basis points at which the shape information is accumulated during inference. Notice that as there are two input arguments of the radial function, the basis points are required to cover a two-dimensional space at a sufficient density to be able to capture the characteristics of the object shape. Also note that the computational load and the memory storage scale with the number of basis points since they are included in the state vector and updated at each recursion. A naive attempt to utilize fewer basis points for more efficient implementation will naturally result in a degraded representational power, potentially missing salient features of the 3D extent which might in turn deteriorate tracking accuracy.

In this section, we will seek for an alternative algorithm with improved computational properties. This second approach essentially retains the basic structure of the previous one; however, it fundamentally differs in the description of the object shape. In particular, multiple projections of the object are exploited to express the original 3D extent. Accordingly, the problem is reformulated as tracking the object while simultaneously learning the contours of its projections. This will eventually enable us to radically lower the number of basis points without compromising the representational power. The next section presents the alternative extent model in details.

3.7.1 Projection Model

It is a long-standing idea to exploit projections, silhouettes or images for expressing the corresponding 3D shape, [67, 68]. Being inspired by these methods, we suggest to model the object extent using projections onto several planes. Fig. 3.3 illustrates the idea for an example object with cone shape. In this case, the object is projected onto three orthogonal planes and the contours of these projections are essentially utilized to represent the original 3D shape. In this exposition, we assume that three orthogonal projections can sufficiently approximate the 3D shape; however, the number of projections can be increased to be able to generalize to a broader class of objects. For a systematic discussion on the objects which are exactly reconstructable from projections and the minimum number of projections necessary for reconstructing such objects, interested readers can refer to [69].

The contour of each projection can be described by a radial function in polar coordinates, i.e., $r = f(\theta)$, as shown in Fig. 3.3. The radial function maps the polar angle, θ, to the radial distance, r, between the projection center and the contour. Notice that having only one input argument, this function can possibly be approximated by a less number of basis points leading to a tracking algorithm demanding less computational sources.

The rest of the derivation closely follows the first algorithm. The unknown radial function on each projection plane is modeled by a GP, i.e., $f(\theta) \sim \mathcal{GP}(\mu(\theta), k(\theta, \theta'))$, whose mean function is taken to be constant $\mu(\theta) = \mu_r$, and the covariance function is defined as

$$k(\theta, \theta') = \sigma_f^2 e^{-\frac{2\sin^2\left(\frac{\theta-\theta'}{2}\right)}{l^2}} + \sigma_r^2. \tag{3.47}$$

Notice that the sin$(\cdot)$ term in (3.47) is used to induce a periodic covariance function, which in turn assures the periodicity of the radial function $f(\cdot)$ described by the given GP model, [26].

Further Discussion: Expressing the 3D shape in terms of a collection of projection contours enables us to introduce separate probabilistic models for each contour to account for application-specific knowledge about the objects. For example, many targets in driving environments, such as cars, vans and bicycles, possess a common characteristic in their projections onto the ground plane. In particular, the radial function describing the corresponding projection contour appears to be periodic with π. In this case, to encode this information into the GP model, the covariance function can be designed as

$$k\left(\theta, \theta'\right) = \sigma_f^2 e^{-\frac{\sin^2(\theta-\theta')}{2l^2}} + \sigma_r^2. \tag{3.48}$$

As the covariance function is periodic with π, the learned contours will comply with

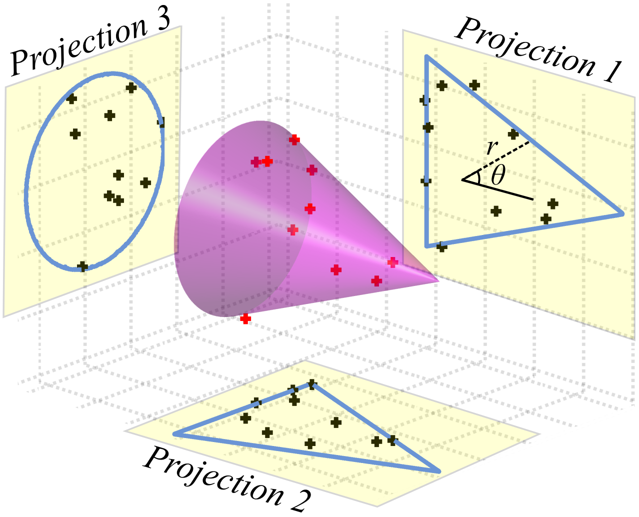


Figure 3.3: Illustration of a cone-shaped object and the corresponding projection contours on three orthogonal planes. Point cloud measurements and their projections are shown by red and black plus signs, respectively.

the actual characteristics of the projections as intended.

3.7.2 State-Space Model

In this subsection, the state-space model relying on the extent description obtained by projection contours is to be constructed. The state vector is defined as $\mathbf{x}_k \triangleq \left[\bar{\mathbf{x}}_k^\top \ \mathbf{f}_k^\top\right]^\top$ where $\bar{\mathbf{x}}_k$ includes the object kinematics and the extent is indicated by $\mathbf{f}_k \triangleq \left[\mathbf{f}_k^{1\top} \ \mathbf{f}_k^{2\top} \ \mathbf{f}_k^{3\top}\right]^\top$ as a collection of the projection contours. More specifically, $\mathbf{f}_k^j$ is the parameterized description of the GP model for the radial function specifying the contour of the projection on the j^{th} plane. $\mathbf{c}_k$ is the center of the 3D object and $\mathbf{v}_k$ stands for the velocity of the center; $\mathbf{q}_k$ is the unit quaternion vector.

3.7.2.1 Measurement Model

The measurement model makes use of the local and global coordinate frames as defined earlier. While the object motion is tracked in the global frame, the shape is described in the local frame. In particular, the projection planes are fixed to the local frame so that the projections of the object onto the planes are kept unchanged at any time. This allows to accumulate extent information over these planes by learning the latent contours of the projections.

Consider the 3D point cloud measurements acquired at time k, $\{\mathbf{m}_{k,i}\}_{i=1}^{n_k}$. Firstly, each measurement $\mathbf{m}_{k,l}$ is transformed into the local frame to obtain $\mathbf{m}_{k,l}^{L}$ by (3.39). Thereafter, these local measurements are to be projected onto each plane to establish a relation between the projected measurements and the projection contour on the corresponding plane. As an example, let $\mathbf{m}_{k,l}^{L}$ be projected onto the j^{th} plane by

$$\mathbf{m}_{k,l}^{j} = P_j \, \mathbf{m}_{k,l}^{L} \, , \tag{3.49}$$

where $\mathbf{m}_{k,l}^{j} \in \mathbb{R}^2$ denotes the projection of $\mathbf{m}_{k,l}^{L}$, and $P_j \in \mathbb{R}^{2x3}$ is the projection matrix.

By assuming that the projections of the measurements are originated from a star convex extent model [26], [70], the projected measurement can be described as

$$\mathbf{m}_{k,l}^{j} = \mathbf{p}_{k,l}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l}) f^{j}(\theta_{k,l}(\mathbf{c}_k,\mathbf{q}_k, \mathbf{m}_{k,l})) + \bar{\mathbf{e}}_{k,l}, \; \bar{\mathbf{e}}_{k,l} \sim \mathcal{N}(0, \bar{R}), \tag{3.50}$$

where $f^{j}(\cdot)$ is the radial function expressing the contour of the projection on the j^{th} plane; $\mathbf{p}_{k,l}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l})$ is the unit-length vector pointing from the projection center towards the measurement; and $\bar{\mathbf{e}}_{k,l}$ is the Gaussian measurement noise with covariance matrix $\bar{R}$. Notice that unlike (3.41), the center position is not superposed in (3.50) as the projection is specified to be centered at the origin of the corresponding plane.

In (3.50), the underlying source of $\mathbf{m}_{k,l}^{j}$ is actually unknown. Therefore, we resort to an approximate approach and formulate the expressions on the right hand side of the measurement model as functions of the kinematic variables and the measurement $\mathbf{m}_{k,l}$ itself.

The polar angle $\theta_{k,l}$ associated with the projected measurement can be computed as

$$\theta_{k,l}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l}) = \angle \mathbf{m}_{k,l}^{j} \, . \tag{3.51}$$

Besides, the unit-length vector $\mathbf{p}_{k,l}$ is obtained by

$$\mathbf{p}_{k,l}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l}) = \frac{\mathbf{m}_{k,l}^{j}}{\|\mathbf{m}_{k,l}^{j}\|} \, . \tag{3.52}$$

The next step is to plug the GP representation for the radial function into (3.50) as

$$\mathbf{m}^j_{k,l} = \tilde{H}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l})\, \mathbf{f}^j_k + \tilde{\mathbf{e}}_{k,l},\ \tilde{\mathbf{e}}_{k,l} \sim \mathcal{N}(0, \tilde{R}_{k,l}), \tag{3.53a}$$

where

$$\begin{aligned}
\tilde{H}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l}) &= \mathbf{p}_{k,l} H^{\mathbf{f}}(\theta_{k,l}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l}))\ , && (3.53\text{b})\\
\tilde{\mathbf{e}}_{k,l} &= \mathbf{p}_{k,l} e^{\mathbf{f}}_{k,l} + \bar{\mathbf{e}}_{k,l}\ , && (3.53\text{c})\\
\tilde{R}_{k,l} &= \mathbf{p}_{k,l} R^{\mathbf{f}}_{k,l} \mathbf{p}^{\top}_{k,l} + \bar{R}\ , && (3.53\text{d})\\
\mathbf{p}_{k,l} &= \mathbf{p}_{k,l}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l})\ , && (3.53\text{e})\\
R^{\mathbf{f}}_{k,l} &= R^{\mathbf{f}}\left(\theta_{k,l}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l})\right)\ . && (3.53\text{f})
\end{aligned}$$

As before, the additive noise term $\tilde{\mathbf{e}}_{k,l}$ in (3.53a) is actually a function of the unknown state vector, and it is not necessarily Gaussian. Again, we ignore this dependence and assume a Gaussian density so that the resulting approximate model lets us use an EKF for inference.

The projected measurements are not necessarily located on the contour, instead some of them may fall within the interior of the projection area as depicted in Fig. 3.3. Accounting for this observation, the measurement model is modified as

$$\mathbf{m}^j_{k,l} = s_{k,l} \tilde{H}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l})\, \mathbf{f}^j_k + \tilde{\mathbf{e}}_{k,l}\ , \tag{3.54}$$

where $s_{k,l} \in [0, 1]$ is a random scaling factor. We approximate s as a Gaussian random variable, i.e., $s_{k,l} \sim \mathcal{N}(\mu_s, \sigma^2_s)$, [12], [26], [70], since an EKF will be employed for inference. In an ideal case, a particular angle dependent one-dimensional probability distribution for the random scaling factor should be chosen depending on the object-sensor geometry, object's shape and the projection planes.

Considering the characteristics of the scaling factor, the measurement model can be rewritten as

$$\begin{aligned}
\mathbf{m}^j_{k,l} &= \underbrace{\mu_s \tilde{H}_{k,l}\, \mathbf{f}^j_k}_{=\tilde{\mathbf{h}}^j(\mathbf{x}_k, \mathbf{m}_{k,l})} + \underbrace{(s_{k,l} - \mu_s)\tilde{H}_{k,l}\, \mathbf{f}^j_k + \tilde{\mathbf{e}}_{k,l}}_{=\mathbf{e}^j_{k,l}}\\
&= \tilde{\mathbf{h}}^j(\mathbf{x}_k, \mathbf{m}_{k,l}) + \mathbf{e}^j_{k,l}, \quad \mathbf{e}^j_{k,l} \sim \mathcal{N}(0, R^j_{k,l})\ , && (3.55\text{a})
\end{aligned}$$

where

$$R^j_{k,l} = \sigma_s^2 \tilde{H}_{k,l}\, \mathbf{f}^j_k\, {\mathbf{f}^j_k}^\top \tilde{H}^\top_{k,l} + \tilde{R}_{k,l} \;, \tag{3.55b}$$

$$\tilde{H}_{k,l} = \tilde{H}(\mathbf{c}_k, \mathbf{q}_k, \mathbf{m}_{k,l}) \;. \tag{3.55c}$$

Then, the expression for the projected measurement is substituted into this equation as

$$P_j R^L_G(\mathbf{q}_k)(\mathbf{m}_{k,l} - \mathbf{c}_k) = \tilde{\mathbf{h}}^j(\mathbf{x}_k, \mathbf{m}_{k,l}) + \mathbf{e}^j_{k,l} \;. \tag{3.56}$$

Finally, collecting the terms on one side of the equation, we end up with the following implicit measurement model with additive Gaussian measurement noise.

$$\begin{aligned}\mathbf{0} &= \underbrace{-P_j R^L_G(\mathbf{q}_k)(\mathbf{m}_{k,l} - \mathbf{c}_k) + \tilde{\mathbf{h}}^j(\mathbf{x}_k, \mathbf{m}_{k,l})}_{=\mathbf{h}^j(\mathbf{x}_k,\mathbf{m}_{k,l})} + \mathbf{e}^j_{k,l} \\ &= \mathbf{h}^j(\mathbf{x}_k, \mathbf{m}_{k,l}) + \mathbf{e}^j_{k,l} \end{aligned} \tag{3.57}$$

This measurement model together with the process model introduced in Section 3.5.1 establishes the state-space model.

3.7.3 Inference

Similar to the former case, an EKF is employed to realize recursive inference. To process all measurements instantaneously at the update phase of the filter, the complete measurement equation is written as

$$\mathbf{0} = \mathbf{h}(\mathbf{x}_k, \mathbf{m}_k) + \mathbf{e}_k, \quad \mathbf{e}_k \sim \mathcal{N}(0, R_k), \tag{3.58a}$$

where

$$\mathbf{h}(\mathbf{x}_k, \mathbf{m}_k) = \left[{\mathbf{h}^1_k}^\top,\ {\mathbf{h}^2_k}^\top,\ {\mathbf{h}^3_k}^\top\right]^\top , \tag{3.58b}$$

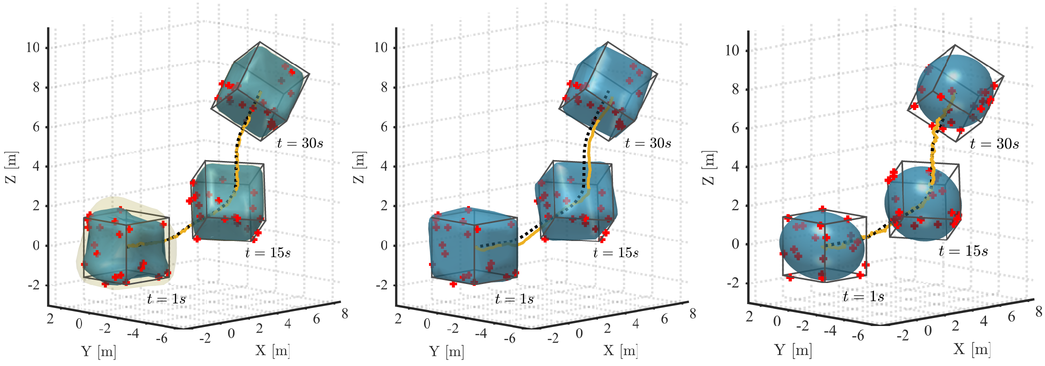
$$\mathbf{m}_k = \left[\mathbf{m}^\top_{k,1},\ \ldots,\ \mathbf{m}^\top_{k,n_k}\right]^\top , \tag{3.58c}$$

$$\mathbf{h}^j_k = \mathbf{h}^j(\mathbf{x}_k, \mathbf{m}_k) = \left[{\mathbf{h}^j}^\top_{k,1},\ \ldots, {\mathbf{h}^j}^\top_{k,n_k}\right]^\top , \tag{3.58d}$$

$$\mathbf{h}^j_{k,l} = \mathbf{h}^j(\mathbf{x}_k, \mathbf{m}_{k,l}) \;, \tag{3.58e}$$

$$R_k = \text{diag}\left[R^1_k,\ R^2_k,\ R^3_k\right] , \tag{3.58f}$$

$$R^j_k = \text{diag}\left[R^j_{k,1},\ \ldots, R^j_{k,n_k}\right] \text{ for } j \in \{1, 2, 3\}. \tag{3.58g}$$



(a) GPEOT

(b) GPEOT-P

(c) RM

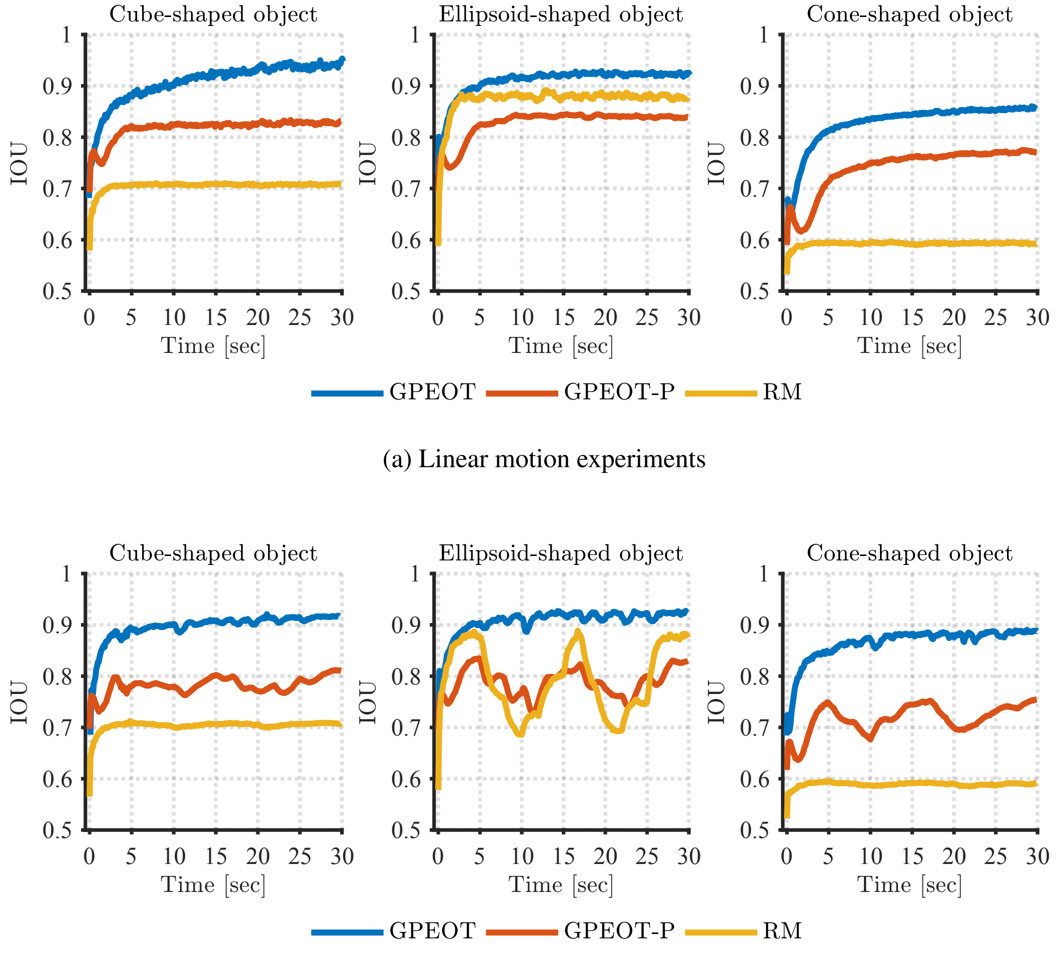
Figure 3.4: Typical results for the cube-shaped object during the complex maneuvering experiment. (Blue surface and black box visualize the estimated and the true extent of the object, respectively. In Fig. (a) yellow surface indicates the confidence interval of one standard deviation. Red plus signs are the point measurements. Solid yellow and dashed black curves are the estimated and true trajectory, respectively.)

3.8 Results

In this section, the performance of the proposed algorithms is evaluated on both simulated and real measurements in Section 3.8.1 and 3.8.2, respectively. To be able to present the results in a comparative manner, we also regard a standard random matrix-based extended object tracker [9], denoted as RM. Describing the extent by an ellipsoid, the RM model has proven to be extremely robust for a wide range of scenarios. Therefore, it serves as a solid basis to assess the tracking performance of the suggested models. Throughout this section, we will refer to the first proposed method as ‘GPEOT’ (short for GP-based extended object tracker), while ‘GPEOT-P’ will stand for the second approach considering the projections.

3.8.1 Experiments with Simulated Measurements

To demonstrate the performance of the algorithms, various simulation experiments are conducted. Section 3.8.1.1 examines the setting where the point cloud measurements are simulated in Matlab for several dynamic objects with basic shapes. In this



(a) Linear motion experiments

(b) Complex maneuvering experiments

Figure 3.5: Intersection-Over-Union (IOU) plots. (The results are averaged over 100 MC runs.)

case, the measurements are randomly sampled from the objects' surfaces. In Section 3.8.1.2, we make use of a specialized sensor simulation environment, namely Blensor, [71], that generates measurements for realistic sensor and vehicle models.

3.8.1.1 Matlab Simulations

In this subsection, the algorithms process point cloud measurements which are generated from random sources on the object surface and perturbed by additive Gaussian noise. Three different-shaped objects, e.g., cube, ellipsoid and cone, are tracked during the experiments. The dimensions of the objects are as follows: the length of the edge of the cube is 3 m, the semi-axes of the ellipsoid are (2.5, 1, 1) m in length, and the base radius and height of the cone are 1.5 m and 4 m, respectively.

The overall performance is evaluated based on the Intersection-Over-Union (IOU) measure given by

$$\text{IOU}(S_{true}, \hat{S}) = \frac{\text{volume}(S_{true} \cap \hat{S})}{\text{volume}(S_{true} \cup \hat{S})}, \tag{3.59}$$

where S_{true} is the true object shape, and $\hat{S}$ stands for the estimate. Notice that IOU simultaneously accounts for the quality of the estimates of the kinematics and the extent. In other words, an algorithm needs to produce accurate tracking outputs together with precise shape description to attain high IOU scores. Also note that in our discussion we deliberately exclude the RMSE measure for the position estimates since the suggested shape models do not imply a unique center definition; instead, different center positions with compatible radial functions can accurately represent the same object.

We also compute the root mean squared error (RMSE) of the object velocity, defined as

$$\text{RMSE}(\hat{\mathbf{v}}_k, \mathbf{v}_k) = \sqrt{\frac{1}{N}\sum_{k=1}^{N} |\hat{\mathbf{v}}_k - \mathbf{v}_k|^2}, \tag{3.60}$$

where $\hat{\mathbf{v}}_k$ and $\mathbf{v}_k$ indicate the estimated and the true velocity vector at time k, respectively; $|\cdot|$ indicates the Euclidean norm.

Two different scenarios are studied in the simulations: a linear motion and a complex maneuver. In the first case, objects move along a linear trajectory at a constant speed of 10 m/s. In the second experiment, the object follows a curved path while performing combined rotations around different axes. Throughout the trajectory, the linear speed is kept constant at 0.5 m/s. At each instant, 20 point measurements are originated from random sources which are sampled from a uniform distribution defined over the object surface. Each point measurement is perturbed by i.i.d. Gaussian noise with covariance $0.1^2 I_3$. The measurements are produced at 10 Hz, hence the sampling time of all algorithms is set to $T = 0.1$. We should note that the simulated measurements used in this section do not follow the actual characteristics of

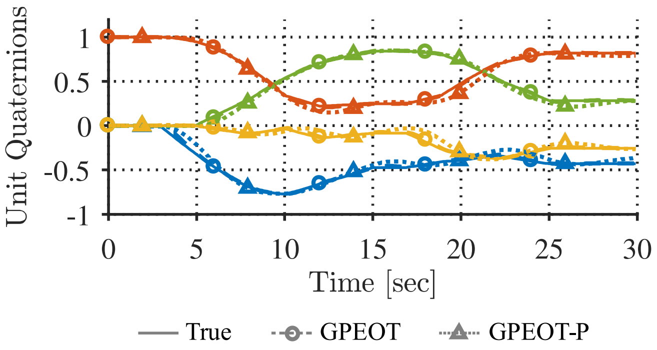


Figure 3.6: True and estimated unit quaternions for the cube-shaped object during the complex maneuvering experiment. (The estimates are averaged over 100 MC runs. Color code is blue: q_0, green: q_1, yellow: q_2, red: q_3.)

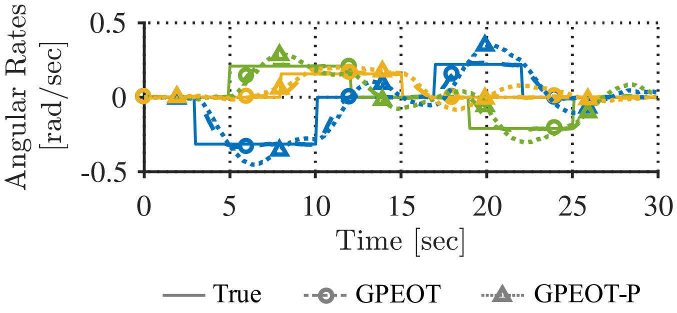


Figure 3.7: True and estimated angular rates for the cube-shaped object during the complex maneuvering experiment. (The estimates are averaged over 100 MC runs. Color code is blue: ω_x, green: ω_y, yellow: ω_z.)

range scanners, which partially delineate the object's surface due to self-occlusions and generate sparse measurements with increased distance between the sensor and the object. We will investigate experiments regarding data collected by real sensors in Section 3.8.2.

For GPEOT, the process noise standard deviations are set to $\sigma_c = 0.1$ and $\sigma_\alpha = 0.1$, and $\lambda = 0.99$ is used for the extent dynamics; the hyper-parameters of the GP model are set to $\mu_r = 0$, $\sigma_f = 1$, $\sigma_r = 0.2$, $l = \pi/8$; $\bar{R} = 0.1^2 I_3$ is used for the measurement noise variance; and the extent is represented by 642 basis points which are evenly spaced with respect to their spherical angles. For GOEOT-P, the process noise standard deviations are set to $\sigma_c = 0.1$ and $\sigma_\alpha = 0.4$, and $\lambda = 0.99$ is used for the extent dynamics; the hyper-parameters of the GP model are set to $\mu_r = 0$, $\sigma_f = 1$, $\sigma_r = 0.2$, $l = \pi/5$; $\bar{R} = 0.1^2 I_2$ is used for the projected measurement noise variance; each projection contour is represented by 50 basis points which are equidistantly located in $[0, 2\pi]$; and the parameters of the scaling factor are set to $\mu_s = \frac{5}{6}$ and $\sigma_s^2 = \frac{1}{18}$. The projection planes are selected to be the xy, xz and yz planes, hence the corresponding projection matrices are $P_1 = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right]$, $P_2 = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right]$, $P_3 = \left[\begin{smallmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right]$. All algorithms use the same prior distribution for the kinematics. The prior distribution for the extent state is directly specified by the corresponding GP model.

The RM model assumes that the measurements have a normal spread, which is specified by the extent state, and hence they are not confined to be originated by the object's surface. In this regard, the method can be considered to be relatively less suitable for sensors like LIDARs, where the measurements are generated exclusively from the surface. The scaling factor included in the RM model is of practical value for such cases as it accounts for the discrepancy between the sensor's actual behavior and the mentioned assumption, [9]. To obtain a competent performance, we manually optimized the parameters of the RM model: The scaling factor and the extension time constant are set to 1/3 and 1, respectively.

Due to page limitations, we hereby present some instances of our findings as representative examples. Typical results for the cube-shaped object performing the complex maneuver are illustrated in Fig. 3.4. Remember that GPEOT-P originally estimates the latent shape by learning the associated projection contours; therefore, to be able to visualize and interpret the results in 3D, we implemented a simple 3D reconstruction algorithm. The algorithm basically starts from a conservative estimate of the underlying 3D volume and refines the estimate by carving out the sections that are inconsistent with the projections. Fig. 3.4(b) exhibits the reconstructed shapes as estimates.

We examine three differently shaped objects (cube, ellipsoid and cone) for the two motion patterns (linear motion and complex maneuver). For each method, all experiments are repeated 100 times with random realizations of the measurement noise and the measurement origins. Table 3.1 reports the averaged RMSE values of the object velocity computed for all simulation experiments. For all experiments, GPEOT and GPEOT-P show superior performance compared to the RM model in terms of velocity estimation. Fig. 3.5 exhibits the IOU results obtained by averaging the Monte Carlo (MC) runs. GPEOT and GPEOT-P produce successful results for all three shapes while the RM model shows satisfactory performance only for the ellipsoid object. It is an expected finding since both of the GP-based approaches are flexible methods to represent any arbitrary star-convex shape, whereas RM essentially models the underlying shape by an ellipsoid. Additionally, the proposed algorithms are shown to be robust enough to handle the model mismatch in kinematics occurring in the complex maneuvering experiment where the constant velocity model is no longer valid for this motion pattern. A particular reason for their robustness is that they can competently track the orientation of the objects (see Figs. 3.6 and 3.7). Finally, GPEOT is observed to outperform the other algorithms with respect to the IOU measure for all cases.

Computation time: Both of the proposed algorithms are basically realized by an EKF, hence the estimates are recursively updated using newly available measurements at each time step. Therefore, the computational requirements do not increase over time and are basically determined by the size of the state vector and the number of the measurements. The state dimension in GPEOT is $\dim(\mathbf{x}_k) = \dim(\mathbf{c}_k) + \dim(\mathbf{v}_k) + \dim(\mathbf{a}_k) + \dim(\boldsymbol{\omega}_k) + \dim(\mathbf{f}_k) = 654$, and in GPEOT-P, it is $\dim(\mathbf{x}_k) = \dim(\mathbf{c}_k) + \dim(\mathbf{v}_k) + \dim(\mathbf{a}_k) + \dim(\boldsymbol{\omega}_k) + \dim(\mathbf{f}_k^1) + \dim(\mathbf{f}_k^2) + \dim(\mathbf{f}_k^3) = 162$. We utilize a naive implementation of EKF for each method without exploiting any code optimization methods. The partial derivatives used in the measurement update

Table 3.1: Root Mean Squared Error (RMSE) of the Object Velocity [ms^{-1}]. (Results are averaged over 100 MC runs.)

	Linear Motion			**Complex Maneuver**		
	Cube	**Ellipsoid**	**Cone**	**Cube**	**Ellipsoid**	**Cone**
GPEOT	0.124	0.125	0.150	0.112	0.120	0.124
GPEOT-P	0.162	0.184	0.200	0.203	0.184	0.192
RMM	0.323	0.239	0.316	0.332	0.245	0.322

phase of the filter are computed numerically. All simulations are conducted in Matlab 2017a on a standard laptop with Intel Core i7-6700HQ 2.60 Hz CPU using 16 GB of RAM. Average computation time for an update is recorded as 37.3 ms for GPEOT, 8.2 ms for GPEOT-P and 0.2 ms for RM model.

An Alternative Process Model for the Extent: To investigate the impact of the extent process model on the performance, we consider an alternative model, which was proposed in [26]:

$$\mathbf{f}_{k+1} = F^{\mathbf{f}}\mathbf{f}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q^{\mathbf{f}}) \,, \tag{3.61a}$$

where

$$F^{\mathbf{f}} = e^{-\alpha T} I, \quad Q^{\mathbf{f}} = (1 - e^{-2\alpha T}) K(\mathbf{u}^{\mathbf{f}}, \mathbf{u}^{\mathbf{f}}) \,. \tag{3.61b}$$

α can be considered as a forgetting factor. Based on this model, we implemented another version of both GPEOT and GPEOT-P for $\alpha = 0.0001$. The same simulation experiments are conducted with this implementation repeating 100 MC runs. Tables 3.2 and 3.3 present the findings by the mean of the IOU measure for the linear motion and the complex manuevering experiment, respectively. The results suggest that the new implementation also achieves successful performance for all cases, and the alternative process model of the extent does not lead to a significant difference with respect the the IOU measure.

Effect of Center Initialization: As mentioned earlier, the proposed extent models do not rely on a unique definition of the center point. Instead, different points within the object extent can be specified to be the center position, and together with compatible

Table 3.2: Mean of the Intersection-over-Union (IOU) Values for the Linear Motion Experiment. (Results are averaged over 100 MC runs.)

	Process Model for the Extent	**Object Shape**		
		Cube	**Ellipsoid**	**Cone**
GPEOT	Maximum Entropy Model	0.908	0.910	0.824
	Forgetting Factor Model	0.906	0.914	0.832
GPEOT-P	Maximum Entropy Model	0.818	0.829	0.741
	Forgetting Factor Model	0.816	0.826	0.737

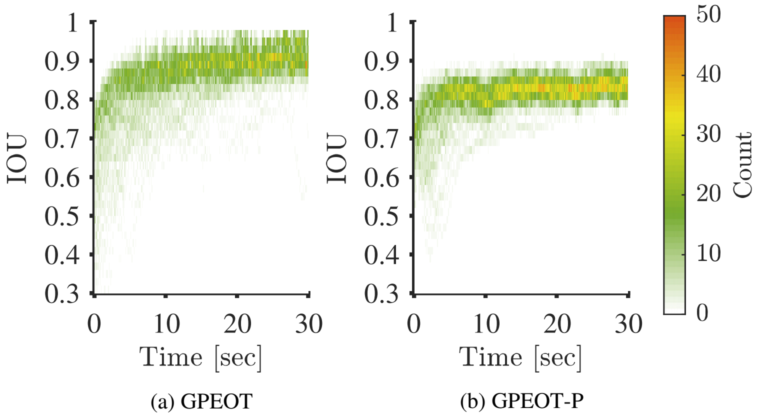
radial functions, these can describe precisely the same object. Therefore, a possible area for future research would be to develop mechanisms to initialize the center point that maximize the performance of the suggested trackers.

In this regard, we examine the sensitivity of the proposed algorithms to the initialization of the center point. We consider the linear motion experiment for the cube-shaped object. The simulation is repeated for 100 times, and at each run, the initial center point is sampled from a uniform distribution defined within the entire volume of the object. The results are illustrated by the histograms that visualize the complete distribution of the IOU values at each time instant of the scenario in Fig. 3.8. Both of the algorithms are shown to be robust against center initialization, as they consistently converge to high IOU values with decreasing variance throughout the experiment.

Effect of Number of Measurements per Frame: In this subsection, we investigate how the performance of the proposed methods depends on the number of available measurements per frame. To this end, we consider the linear motion experiment for the cube-shaped object. We repeat the experiment for the following number of measurements per frame: $\{3, 5, 10, 20, 30\}$. The simulations are run 100 times for each case, where the measurement sources are randomly sampled from the surface and corrupted by independent measurement noise. Fig. 3.9 reports the averaged IOU measure for all experiments. Both methods perform satisfactorily even when the number of measurements is relatively low. Additionally, the performance of the methods regularly improves with the increasing number of measurements, which indicates that they can effectively assimilate information provided by the measurements.

Table 3.3: Mean of the Intersection-over-Union (IOU) Values for the Complex Maneuvering Experiment. (Results are averaged over 100 MC runs.)

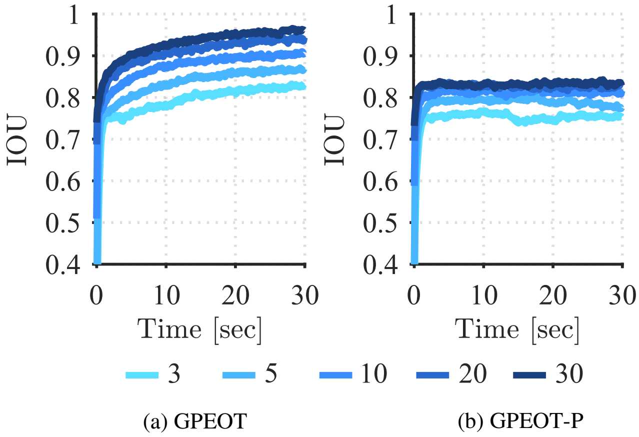
	Process Model for the Extent	**Object Shape**		
		Cube	**Ellipsoid**	**Cone**
GPEOT	Maximum Entropy Model	0.897	0.907	0.866
	Forgetting Factor Model	0.896	0.909	0.865
GPEOT-P	Maximum Entropy Model	0.782	0.788	0.718
	Forgetting Factor Model	0.786	0.786	0.715



(a) GPEOT

(b) GPEOT-P

Figure 3.8: Histogram plots of the Intersection-over-Union (IOU) measure calculated for 100 MC runs of the linear motion experiment with randomly initialized center points.



(a) GPEOT

(b) GPEOT-P

Figure 3.9: Intersection-over-Union (IOU) plots obtained for different number of available measurements per frame. (The results are averaged over 100 MC runs.)

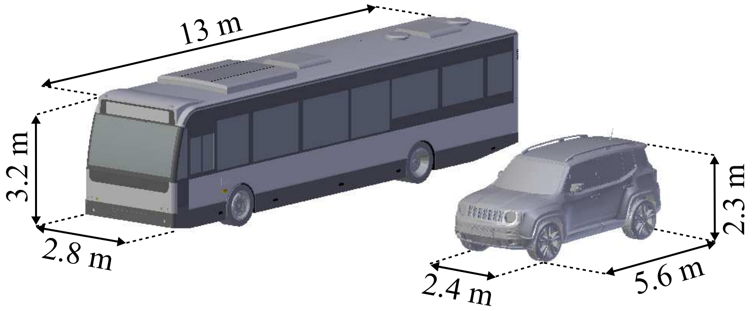


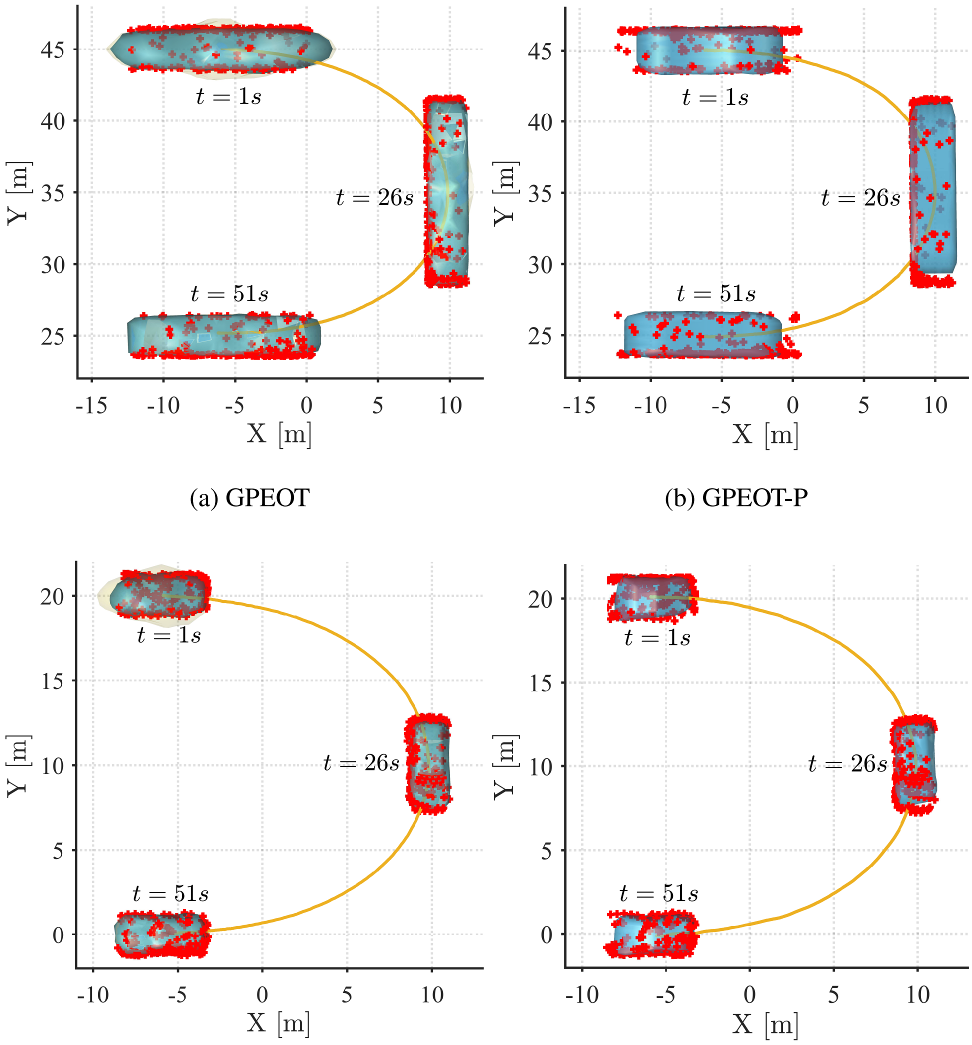
Figure 3.10: Realistic vehicle models utilized in the Blensor experiments.

3.8.1.2 Blensor Simulations

To qualify the representational power of the suggested algorithms, additional experiments are conducted in Blensor, which is a high fidelity sensor simulation environment. In these experiments, we consider realistic models of two different types of vehicles, namely a bus and a jeep, which are depicted in Fig. 3.10. In the scenario, each vehicle makes a u-turn while being observed by two Velodyne HDL-64E2 LIDAR sensors. The parameters of the algorithms are kept the same as in the previous subsection with two exceptions: the covariance matrix of rotational process noise is set to $\Sigma_\alpha = \text{diag}(0, 0, \sigma_\alpha^2)$ to reflect that the considered vehicles can only rotate around their yaw-axis, and the length-scale of GPEOT is set to $l = \pi/12$.

The overview of the tracking outputs is shown in Fig. 3.11. Both of the proposed methods can successfully track the two different vehicles. Furthermore, the shape estimates obtained at the last instant of the experiments are demonstrated by some close-up views in Fig. 3.12. While GPEOT is able to capture a highly detailed representation of the underlying object extent, GPEOT-P achieves a satisfactory but rather rough shape estimate. Besides, GPEOT-P slightly underestimates the size of the object due to the mismatch between the specified and true values of the scaling factor used in the measurement model.

Finally, to further investigate the effect of the number of measurements on the estimation performance, we consider the same Blensor experiment with the jeep vehicle. The employed LIDAR sensors are originally able collect an abundant number of measurements returned from the object of interest, which is well over a thousand for the most sensor scans. For our purposes, we randomly select 10 measurements from the acquired point cloud at each frame and provide them to the algorithms. The resulting extent estimates obtained at the end of the scenario are illustrated in Fig. 3.13. Both of the algorithms estimate the kinematics and the extent satisfactorily during the experiment. In particular, GPEOT can successfully form such a detailed description of the underlying shape with only 10 measurements per frame. In this regard, the proposed method does not require the substantial amount of information generated by the LIDAR sensors; instead, it can effectively perform with significantly fewer measurements.



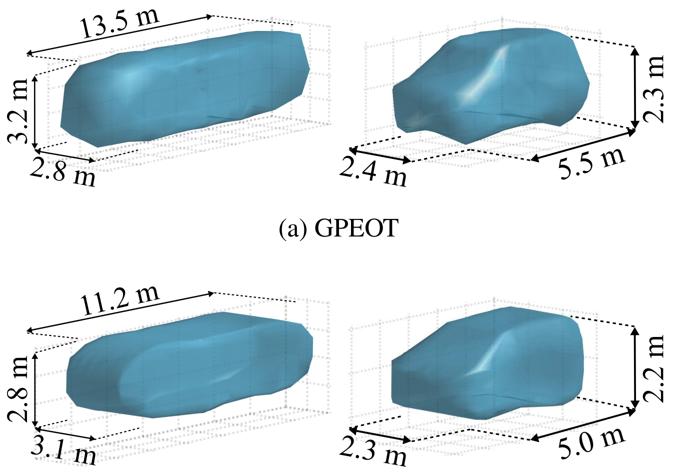
(a) GPEOT

(b) GPEOT-P

(c) GPEOT

(d) GPEOT-P

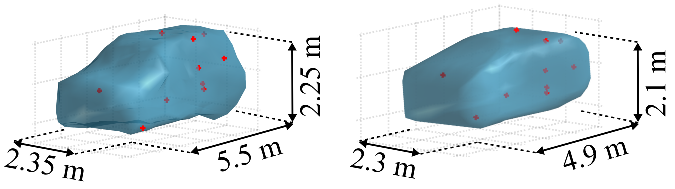
Figure 3.11: Results obtained during Blensor simulations. (In Figs. (a) and (b), the bus is observed by two sensors at (0, 60, -5) and (0, 15, 5). In Figs. (c) and (d), the jeep is observed by two sensors at (0, 30, -5) and (0, -10, 5).)



(a) GPEOT

(b) GPEOT-P

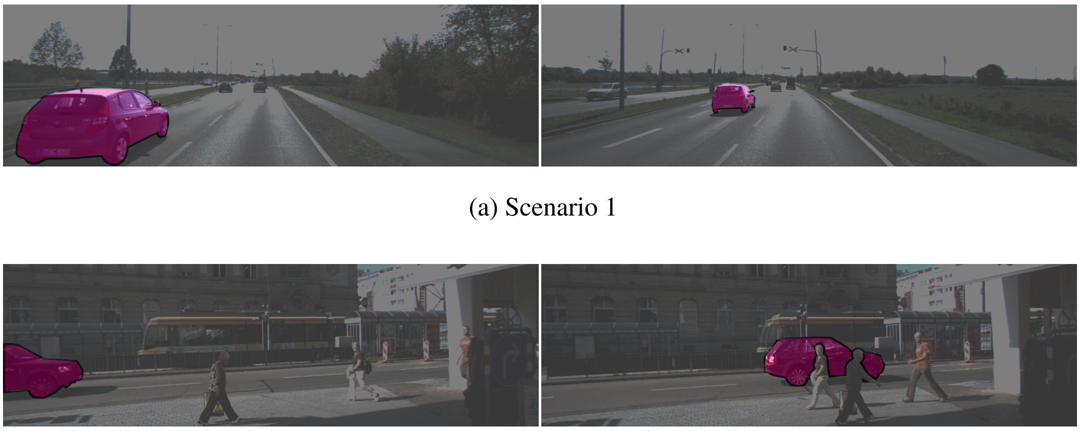
Figure 3.12: Close-up views of the extent estimates obtained at the last instant of the Blensor simulations.



(a) GPEOT

(b) GPEOT-P

Figure 3.13: Close-up views of extent estimates obtained at the last instant of the Blensor experiment, in which only 10 measurements per frame are provided to the algorithms.



(a) Scenario 1

(b) Scenario 2

Figure 3.14: Example views captured by a camera mounted next to the laser scanner on the ego vehicle. Left and right images depict the initial and the intermediate frames of the scenarios, respectively. Note that the highlighted vehicles are tracked by the proposed algorithms using only point cloud measurements.

3.8.2 Experiments with Real Data

In this section, the performance of the algorithms is assessed on real data. To this end, we hereby use the Kitti tracking benchmark, [72]. The benchmark consists of various records of real-world traffic scenarios captured by several sensor modalities mounted on an ego vehicle. We form two scenarios of different vehicles by extracting the corresponding sequences of point measurements acquired by a Velodyne HDL-64E laser scanner. Note that we do not consider the preprocessing of the raw point cloud data, e.g., ground removal, segmentation, association, as it is beyond the scope of this work; instead, the sequences are extracted using the labels provided in the benchmark. The same sets of parameters as in the previous subsections are utilized except the length-scale of GPEOT is set to $l = \pi/14$.

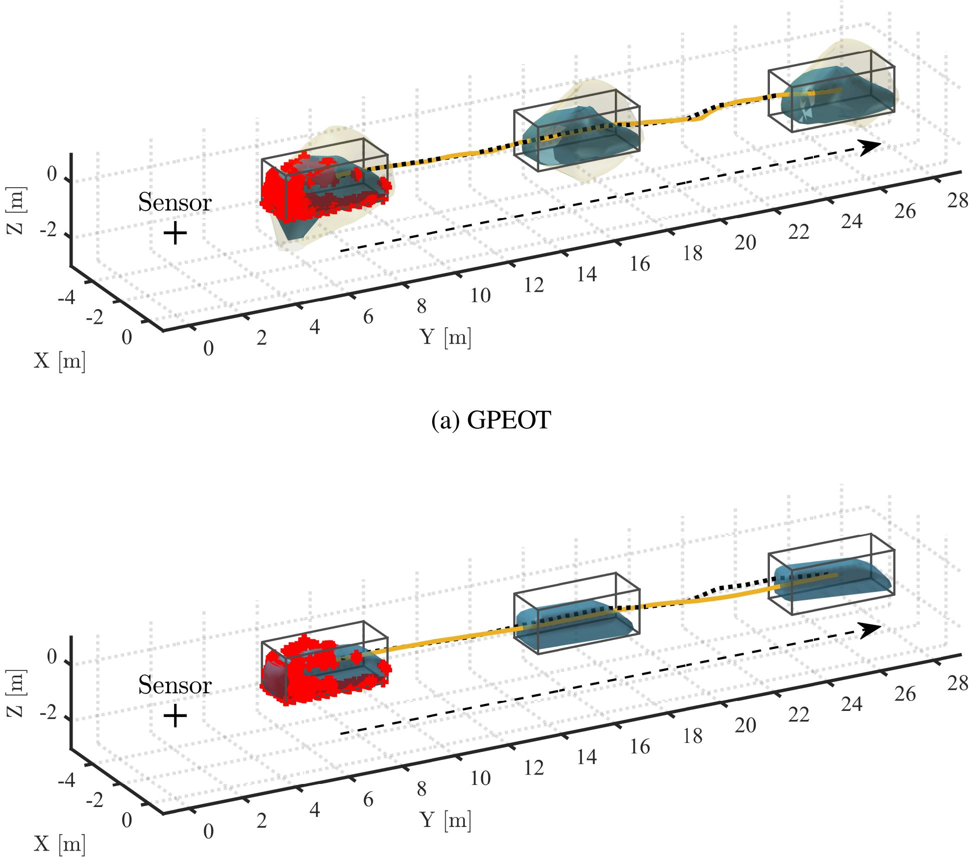
The scenarios are visualized in Fig. 3.14. The first scenario takes place on a highway where the ego and the target vehicle move in the same direction, and the target pulls consistently ahead in time. The dataset provides the ground truth in terms of bounding boxes; hence, to assess the algorithms' performance quantitatively, we fit bounding boxes to the estimated extents and compute the RMSE of the center and

Table 3.4: Root Mean Squared Error (RMSE) of the Center [m] and the Yaw Angle [degree] in the Real Data Experiments.

	Scenario 1		**Scenario 2**	
	Center	**Yaw Angle**	**Center**	**Yaw Angle**
GPEOT	0.21	1.37	0.23	1.40
GPEOT-P	0.27	3.81	0.25	1.90

the yaw angle accordingly. Table 3.4 presents the corresponding results. Fig. 3.15 demonstrates that both of the algorithms accomplish successful tracking. Throughout the experiment, the sensor can only observe the back and right side of the target, thus the uncertainty of the extent on the observed portion decreases in time while a high uncertainty is properly associated with the unobserved section as explicitly shown in Fig. 3.15. The GPEOT-P implementation uses the periodic covariance function in (3.48) for the projection onto the ground plane. Considering solid amount of empirical evidence, this is a reasonable assumption for many targets in driving settings. The implementation inherently assumes that the corresponding projection contour is periodic with π so that the radial function takes exactly same values for $f(\theta)$ and $f(\theta+\pi)$. Accordingly, the reconstructed shape estimates accurately captures the appearance on the unobserved section of the object as seen in Fig. 3.15.

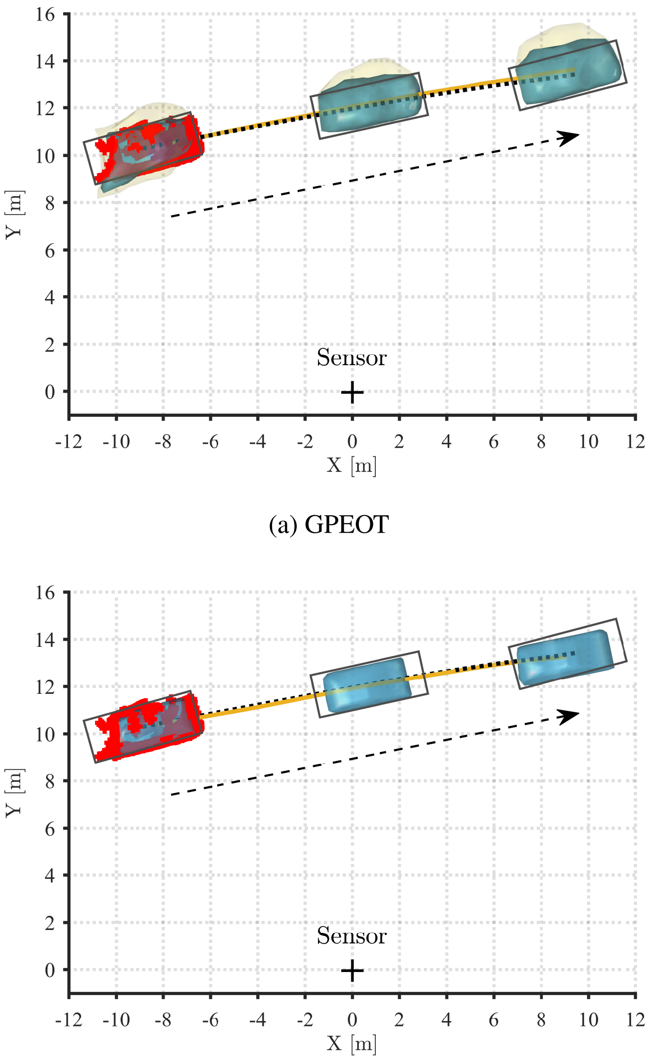
In the second scenario, the ego vehicle waits stationary at a road junction while the target vehicle crosses the street. The experiment imposes two main challenges: First, the target is temporarily occluded by pedestrians and a column of a building; second, there are respectable number of 3D point measurements returned from the driver and the interior structure of the target vehicle. The tracking outputs of the algorithms and the corresponding RMSE values are presented in Fig. 3.16 and Table 3.4, respectively. GPEOT-P makes use of the periodic covariance function for the ground projection as in the previous case. Both of the methods achieve accurate tracking and prove their robustness against occlusions and interior measurements.



(a) GPEOT

(b) GPEOT-P

Figure 3.15: Scenario 1 with real data. (Blue and yellow surfaces indicate the estimated extent and the predicted uncertainty of one standard deviation, respectively. Bounding box denotes the ground truth annotation of the target. Red plus signs plotted for the first frame visualize the measurements. Solid yellow and dashed black curves are the estimated and true trajectory, respectively. Dashed black arrow is the direction of the target.)



(a) GPEOT

(b) GPEOT-P

Figure 3.16: Scenario 2 with real data. (Blue and yellow surfaces indicate the estimated extent and the predicted uncertainty of one standard deviation, respectively. Bounding box denotes the ground truth annotation of the target. Red plus signs plotted for the first frame visualize the measurements. Solid yellow and dashed black curves are the estimated and true trajectory, respectively. Dashed black arrow is the direction of the target.)

3.9 Conclusion and Future Work

A new approach is proposed to contribute to the perception of autonomous systems. The proposed algorithm is capable of processing 3D point cloud data for tracking dynamic objects with unknown shapes. The method can exploit the full potential of the information hidden in, possibly sparse, point cloud measurements by estimating the object's shape simultaneously with its kinematic state including the position, velocity and orientation. The proposed model is flexible to express and learn a large variety of shapes which may co-exist in a surveillance region. An alternative efficient implementation of the method is also derived, which reduces the computational requirements by utilizing plane projections. The algorithms are efficient in the implementation such that an extension to the multi-target tracking framework and real-time applications are possible. The methods provide an analytical expression of the object's shape, and this information can later be used for online identification and classification purposes in the future. The performance of the suggested approach can further be improved by using negative information embedded in the point measurements, considering alternative covariance functions and inference methods, and incorporating dedicated models for the scaling factor in the projections. The methods are applicable to various fields which require accuracy in perception such as the robotics applications where a moving agent is required to navigate in a 3D environment with 3D motion constraints or offline applications such as 3D map extraction.

CHAPTER 4

VARIATIONAL INFERENCE FOR EXTENDED OBJECT TRACKING USING GAUSSIAN PROCESSES

4.1 Introduction

Traditional object tracking is based on the point-source model that assumes an object can generate at most one measurement per scan, [1]. However, recent advances in sensor technology have led to the proliferation of applications where a single object might give rise to multiple measurements per frame. This type of problem is systematically addressed by a well-established line of research, which is referred to as *extended object tracking* (EOT). The objective of these methods is to estimate the object's extent, together with the kinematic variables. There have been developed many successful solutions, which rely on various formulations, e.g., [8, 9, 70, 73, 11, 74, 10, 75, 76, 77, 78, 79]. A comprehensive survey of the existing literature can be found in [3] and [80].

In this chapter, we focus on the Gaussian Process (GP) based EOT methods where the unknown extent is modelled by a GP prior and an approximate inference is performed to obtain its posterior. The resulting trackers estimate the kinematic state of the object while simultaneously learning its unknown extent.

The seminal model [26], here denoted as GPEOT, has been widely adopted by the research community; see, for example, [27, 25, 54, 53, 28, 31, 29, 32, 30]. However, a critical shortcoming of the current GPEOT literature is that the variety of the developed inference techniques is limited, and the resulting trackers typically rely on variants of the Kalman filter in the presence of non-linear, implicit measurement model [31, 29, 32, 30]. Although this approach has been proved to lead to a successful tracking framework, it suffers from the well-known problems of nonlinear Kalman filtering. For instance, poorly predicted state density and high levels of uncertainty in the model might radically degrade the estimation performance. These limitations potentially impede the widespread use of the method in various tracking applications.

In this section of the thesis work, we propose an inference method for the GPEOT model. The suggested method is based on an analytical measurement update, which is derived using the variational Bayes techniques. The resulting algorithm constitutes a recursive tracking filter, which is shown to significantly improve tracking performance and be robust against model uncertainties via comprehensive experiments.

4.2 Proposed Method

GPEOT aims at estimating unknown contours of dynamic objects. The key idea behind the GPEOT approach is to describe the extent of a star-convex[1] object by a radial function $f(\theta)$ which is assumed to have a GP prior, i.e., $f(\theta) \sim \mathcal{GP}(0, k(\theta, \theta'))$, where $k(\theta, \theta')$ is the covariance function, which basically computes the covariance between the function evaluations at angles θ and θ', [26],[42]. To facilitate an efficient, recursive inference mechanism, the GP model is approximated for the function values $\mathbf{f} \triangleq \left[f(\theta_1^{\mathbf{f}}) \dots f(\theta_M^{\mathbf{f}})\right]^\top$ at some basis inputs. The resulting measurement model, which relates the latent variables to a point measurement $\mathbf{y}_{k,l}$, is given in the following [26].

$$\begin{aligned}\mathbf{y}_{k,l} &= \underbrace{\mathbf{c}_k + \tilde{H}_l^{\mathbf{f}}(\mathbf{c}_k, \psi_k)\mathbf{f}_k}_{=\mathbf{h}_{k,l}(\mathbf{x}_k)} + \mathbf{e}_{k,l} \\ &= \mathbf{h}_{k,l}(\mathbf{x}_k) + \mathbf{e}_{k,l},\ \mathbf{e}_{k,l} \sim \mathcal{N}(0, R_{k,l}), \qquad (4.1)\end{aligned}$$

where $\mathbf{c}_k$ is a reference point (referred to as *object center*) within the body of the object; ψ_k denotes the object orientation, $\mathbf{x}_k$ is a unified state vector including both the kinematic and the extent variables, and $\mathbf{e}_{k,l} \sim \mathcal{N}(0, R_{k,l})$ denotes the zero-mean Gaussian measurement noise with covariance $R_{k,l}$. Full details of the variables used in (4.1) is revealed in Appendix A.6.

The representation for the complete measurement set $\{\mathbf{y}_{k,l}\}_{l=1}^{n_k}$ collected at time k is

[1] A set $\mathcal{S}$ is called star-convex if each line segment from the origin to any point in $\mathcal{S}$ is fully contained in $\mathcal{S}$.

formed as

$$\begin{aligned} \mathbf{y}_k &= \left[\mathbf{y}_{k,1}^\top, \ldots, \mathbf{y}_{k,n_k}^\top\right]^\top, R_k = \text{diag}\left[R_{k,1}, \ldots, R_{k,n_k}\right], \\ \mathbf{h}_k(\mathbf{x}_k) &= \left[\mathbf{h}_{k,1}(\mathbf{x}_k)^\top, \ldots, \mathbf{h}_{k,n_k}(\mathbf{x}_k)^\top\right]^\top, \end{aligned} \tag{4.2}$$

where diag$[\cdot]$ returns a block diagonal matrix R_k by placing the matrices $\{R_{k,l}\}_{l=1}^{n_k}$ along the diagonal.

It is possible to construct a state-space model by combining the given measurement equation with a dynamic model of the state vector, and treat the estimation problem in the Bayesian framework.

4.2.1 Variational Measurement Update

Suppose at time k, we have the following predicted density for the kinematic and the extent states

$$p(\bar{\mathbf{x}}_k, \mathbf{f}_k | \mathbf{y}_{1:k-1}) = \mathcal{N}(\bar{\mathbf{x}}_k; \boldsymbol{\mu}_{k|k-1}^x, P_{k|k-1}^x)\mathcal{N}(\mathbf{f}_k; \boldsymbol{\mu}_{k|k-1}^f, P_{k|k-1}^f), \tag{4.3}$$

where $\boldsymbol{\mu}_{k|k-1}$ and $P_{k|k-1}$ denote the predicted mean and covariance, respectively. $\bar{\mathbf{x}}_k$ indicates the kinematic state, which includes the object center $\mathbf{c}_k$, the orientation ψ_k, and the linear and angular velocities $\mathbf{x}_k^*$, i.e., $\bar{\mathbf{x}}_k = \left[\mathbf{c}_k^\top \ \psi_k \ (\mathbf{x}_k^*)^\top\right]^\top$, and $\mathbf{f}_k$ is the extent state. Considering the model in (4.1), the measurement likelihood function can be written as

$$p(\mathbf{y}_k | \bar{\mathbf{x}}_k, \mathbf{f}_k) = \mathcal{N}(\mathbf{y}_k; \mathbf{h}_k(\mathbf{x}_k), R_k), \tag{4.4}$$

where $\mathbf{x}_k$ is the unified state vector including the kinematic and the extent state, i.e., $\mathbf{x}_k = \begin{bmatrix} \bar{\mathbf{x}}_k^\top & \mathbf{f}_k^\top \end{bmatrix}^\top$. In Bayesian filtering, the aim is to obtain the posterior density $p(\bar{\mathbf{x}}_k, \mathbf{f}_k | \mathbf{y}_{1:k})$. Following the classical variational Bayes approach, we approximate the posterior as $p(\bar{\mathbf{x}}_k, \mathbf{f}_k | \mathbf{y}_{1:k}) \approx q(\bar{\mathbf{x}}_k, \mathbf{f}_k) = q_x(\bar{\mathbf{x}}_k) q_f(\mathbf{f}_k)$, where $q_x(\bar{\mathbf{x}}_k)$ and $q_f(\mathbf{f}_k)$ denote the factorized densities for the kinematic and the extent state, respectively.

The variational formulation seeks for the approximate posterior that minimize the Kullback-Leibler (KL) divergence between the true and the approximate posterior, and the solution satisfies the following equations, [41, Ch. 10],

$$\log\ q_x(\bar{\mathbf{x}}_k) = \mathbb{E}_{q_f}[\log\ p(\mathbf{y}_k, \bar{\mathbf{x}}_k, \mathbf{f}_k|\mathbf{y}_{1:k-1})] + c_x, \tag{4.5a}$$

$$\log\ q_f(\mathbf{f}_k) = \mathbb{E}_{q_x}[\log\ p(\mathbf{y}_k, \bar{\mathbf{x}}_k, \mathbf{f}_k|\mathbf{y}_{1:k-1})] + c_f, \tag{4.5b}$$

where c_x and c_f stand for the collection of terms that are not functions of the corresponding factor. The estimated densities are obtained by iteratively computing one of the factors while keeping the other fixed at its most recent estimate, [41, Ch. 10]. These fixed-point iterations are guaranteed to converge to a solution that corresponds to a local minimum of the KL divergence, [81, Sec. 2.5].

To obtain analytical expressions for (4.5), we first need to express the joint density $p(\mathbf{y}_k, \bar{\mathbf{x}}_k, \mathbf{f}_k|\mathbf{y}_{1:k-1})$ as

$$p(\mathbf{y}_k, \bar{\mathbf{x}}_k, \mathbf{f}_k|\mathbf{y}_{1:k-1}) = p(\mathbf{y}_k|\bar{\mathbf{x}}_k, \mathbf{f}_k)p(\bar{\mathbf{x}}_k, \mathbf{f}_k|\mathbf{y}_{1:k-1}). \tag{4.6}$$

Next, we will compute the estimated densities at the $(i+1)^{th}$ iteration, which are denoted by $q_x^{(i+1)}(\bar{\mathbf{x}}_k)$ and $q_f^{(i+1)}(\mathbf{f}_k)$.

4.2.1.1 Computation of $q_x^{(i+1)}(\bar{\mathbf{x}}_k)$

Substituting (4.6) in (4.5a), we can write

$$\log\ q_x^{(i+1)}(\bar{\mathbf{x}}_k) = \mathbb{E}_{q_f^{(i)}}[\log\ p(\mathbf{y}_k|\bar{\mathbf{x}}_k, \mathbf{f}_k)] + \mathbb{E}_{q_f^{(i)}}[\log\ p(\bar{\mathbf{x}}_k|\mathbf{y}_{1:k-1})] + c_x. \tag{4.7}$$

Unfortunately, this expression does not yield a Gaussian density for $q_x^{(i+1)}$ due to the nonlinearity in the mean of $p(\mathbf{y}_k|\bar{\mathbf{x}}_k, \mathbf{f}_k)$. To alleviate this problem, we approximate the measurement model by its first-order Taylor series expansion.

$$\begin{aligned}\mathbf{y}_k &= \mathbf{h}_k(\boldsymbol{\mu}_x^{(i)}, \boldsymbol{\mu}_f^{(i)}) + A^{(i)}\big(\bar{\mathbf{x}}_k - \boldsymbol{\mu}_x^{(i)}\big) + B^{(i)}\big(\mathbf{f}_k - \boldsymbol{\mu}_f^{(i)}\big) + \mathbf{e}_k \\ &= \mathbf{g}^{(i)} + A^{(i)}\bar{\mathbf{x}}_k + B^{(i)}\mathbf{f}_k + \mathbf{e}_k \end{aligned} \tag{4.8a}$$

$$\mathbf{g}^{(i)} \triangleq \mathbf{h}_k(\boldsymbol{\mu}_x^{(i)}, \boldsymbol{\mu}_f^{(i)}) - A^{(i)}\boldsymbol{\mu}_x^{(i)} - B^{(i)}\boldsymbol{\mu}_f^{(i)} \tag{4.8b}$$

$$A^{(i)} \triangleq \frac{\partial \mathbf{h}_k}{\partial \bar{\mathbf{x}}_k}|_{\bar{\mathbf{x}}_k=\boldsymbol{\mu}_x^{(i)}, \mathbf{f}_k=\boldsymbol{\mu}_f^{(i)}}, \quad B^{(i)} \triangleq \frac{\partial \mathbf{h}_k}{\partial \mathbf{f}_k}|_{\bar{\mathbf{x}}_k=\boldsymbol{\mu}_x^{(i)}, \mathbf{f}_k=\boldsymbol{\mu}_f^{(i)}} \tag{4.8c}$$

Note that the Taylor series approximation is performed within every VB iteration which can break the convergence results of the VB. However, we did not observe any divergence tendency of the algorithm in our experiments with simulated and real data.

Consequently, we can write $\mathbb{E}_{q_f^{(i)}}[\log\ p(\mathbf{y}_k|\bar{\mathbf{x}}_k,\mathbf{f}_k)]$ as

$$\begin{aligned}\mathbb{E}_{q_f^{(i)}}[\log\ p(\mathbf{y}_k|\bar{\mathbf{x}}_k,\mathbf{f}_k)] = -0.5\,\big(&\mathbf{y}_k-\mathbf{g}^{(i)}-A^{(i)}\bar{\mathbf{x}}_k)^\top R_k^{-1}(\mathbf{y}_k-\mathbf{g}^{(i)}-A^{(i)}\bar{\mathbf{x}}_k)\\ &-2(\mathbf{y}_k-\mathbf{g}^{(i)}-A^{(i)}\bar{\mathbf{x}}_k)^\top R_k^{-1}B\boldsymbol{\mu}_f^{(i)}\Big)+c_x. \end{aligned}\qquad(4.9)$$

It can be shown that the variational iterate in (4.7) becomes a Gaussian distribution, i.e., $q_x^{(i+1)}(\bar{\mathbf{x}}_k)=\mathcal{N}(\boldsymbol{\mu}_x^{(i+1)},P_x^{(i+1)})$, and its mean and covariance can be computed by a Kalman filter measurement update as follows.

$$\begin{aligned}\boldsymbol{\mu}_x^{(i+1)} &= \boldsymbol{\mu}_{k|k-1}^x + K_k\left(\mathbf{y}_k-\mathbf{g}^{(i)}-B^{(i)}\boldsymbol{\mu}_f^{(i)}-A^{(i)}\boldsymbol{\mu}_{k|k-1}^x\right)\\ P_x^{(i+1)} &= P_{k|k-1}^x - K_kA^{(i)}P_{k|k-1}^x\\ S_k &= A^{(i)}P_{k|k-1}^x(A^{(i)})^\top + R_k\\ K_k &= P_{k|k-1}^x(A^{(i)})^\top(S_k)^{-1}\end{aligned}\qquad(4.10)$$

4.2.1.2 Computation of $q_f^{(i+1)}(\mathbf{f}_k)$

From (4.5b) and (4.6), it follows

$$\log\ q_f^{(i+1)}(\mathbf{f}_k)=\mathbb{E}_{q_x^{(i+1)}}[\log\ p(\mathbf{y}_k|\bar{\mathbf{x}}_k,\mathbf{f}_k)]+\mathbb{E}_{q_x^{(i+1)}}[\log\ p(\mathbf{f}_k|\mathbf{y}_{1:k-1})]+c_f.\qquad(4.11)$$

By using the measurement model in (4.1), we can write

$$\begin{aligned}\mathbb{E}_{q_x^{(i+1)}}[\log\ p(\mathbf{y}_k|\bar{\mathbf{x}}_k,\mathbf{f}_k)] = &-0.5\Bigg(\mathbf{f}_k^\top\underbrace{\overbrace{\mathbb{E}[\tilde{H}(\bar{\mathbf{x}}_k)^\top R_k^{-1}\tilde{H}(\bar{\mathbf{x}}_k)]}^{\triangleq\, g_1(\bar{\mathbf{x}}_k)}}_{\triangleq\,\Delta}\mathbf{f}_k\\ &-2\mathbf{f}_k^\top\underbrace{\overbrace{\mathbb{E}[\tilde{H}(\bar{\mathbf{x}}_k)^\top R_k^{-1}(\mathbf{y}_k-\mathbf{c}_k)]}^{\triangleq\, g_2(\bar{\mathbf{x}}_k)}}_{\triangleq\,\boldsymbol{\delta}}\Bigg)+c_f\end{aligned}\qquad(4.12)$$

The expectations indicated by Δ and $\boldsymbol{\delta}$ include highly nonlinear functions of $\bar{\mathbf{x}}_k$, and do not end up with compact analytical expressions. We employ the unscented transformation [82] to approximately compute these expectations as

$$\Delta\approx\sum_{j=0}^{M}\pi^j g_1(\bar{\mathbf{x}}_k^j),\ \boldsymbol{\delta}\approx\sum_{j=0}^{M}\pi^j g_2(\bar{\mathbf{x}}_k^j),\qquad(4.13)$$

where $\{\bar{\mathbf{x}}_k^j\}_{j=0}^M$ and $\{\pi^j\}_{j=0}^M$ denote the sigma-points and the associated weights, which are explicitly defined in Appendix A.5.

The resulting expression in (4.12) together with the prediction density imply that the variational iterate for the extent is a Gaussian density, and we can analytically compute its mean $\boldsymbol{\mu}_f^{(i+1)}$ and covariance $P_f^{(i+1)}$ as follows

$$\begin{aligned} \boldsymbol{\mu}_f^{(i+1)} &= \left(\Delta + (P^f_{k|k-1})^{-1}\right)^{-1} \left(\boldsymbol{\delta} + (P^f_{k|k-1})^{-1}\boldsymbol{\mu}^f_{k|k-1}\right), \\ P_f^{(i+1)} &= \left(\Delta + (P^f_{k|k-1})^{-1}\right)^{-1}. \end{aligned} \tag{4.14}$$

This concludes the derivation of the variational approximation-based measurement update. This mechanism, combined with a process model of the state vector, establishes an EOT approach; and we will examine the performance of the corresponding algorithm in the following sections.

4.2.2 A Closer Look to a Single Measurement Update

Before proceeding to a detailed performance analysis, we first want to demonstrate the capabilities of the proposed method, denoted as 'VB', and the reference method [26], named 'EKF'. For this purpose, we focus on a single measurement update that is realized for the setting in Fig. 4.1.

In this example, the underlying object has a rectangular extent. Both methods are initialized by the same prior and are provided with the same set of measurements. We compare the resulting posterior distributions. Contrary to EKF, the VB algorithm performs multiple iterations in a single measurement update. Consequently, the posterior obtained by VB is significantly closer to the true posterior compared to that of EKF. This desirable behavior is the primary motivation behind the development of VB, and it will be demonstrated to lead to superior tracking performance in Section 4.3.

4.3 Experimental Results

4.3.1 Experiments with Simulated Data

We conduct various experiments with simulated data considering different motion behaviors and extent models. Measurements are randomly generated from the under-

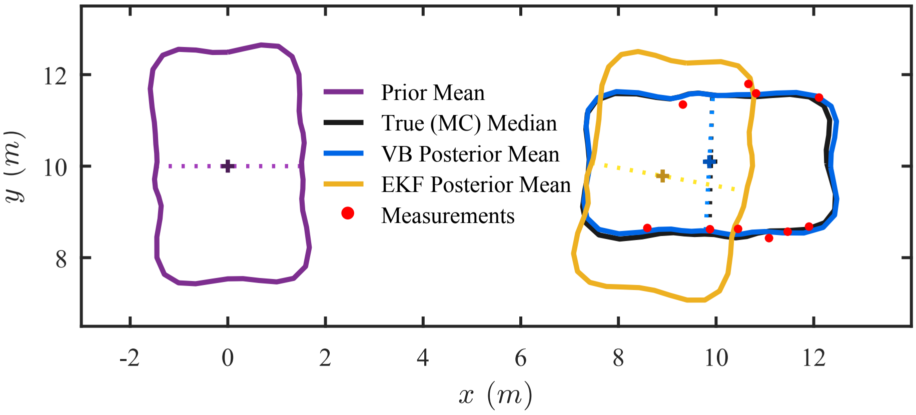


Figure 4.1: Illustration of a single measurement update. We also obtain the median of the true posterior distribution by using 1 million Monte Carlo (MC) samples, which is plotted in black. The number of variational iterations is 10 for VB. (The plus signs and the dashed lines indicate the estimates of the object center and the orientation, respectively.)

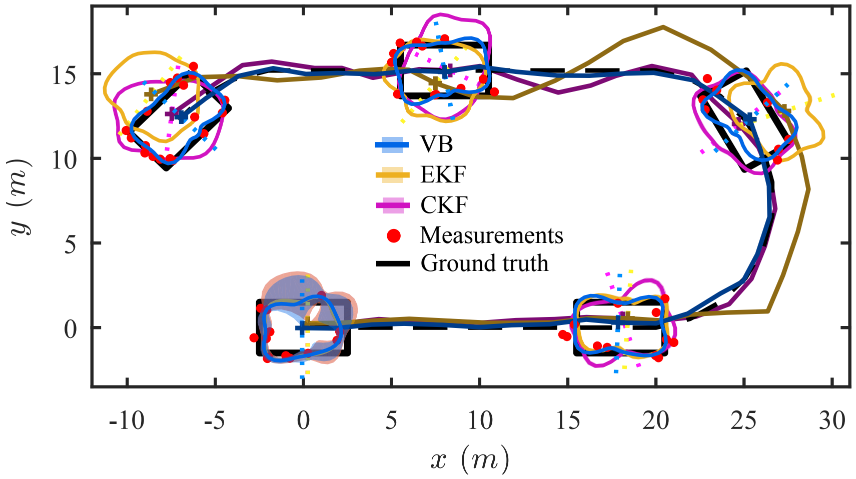


Figure 4.2: A typical realization of S2-Exp2 for a rectangular object.

lying object contour and corrupted by Gaussian noise with covariance $R = 0.3^2 I_2$ m^2. The number of the measurements on each scan is sampled from a Poisson distribution with mean 15.

To assess the performance of the algorithms, we regard the Intersection-Over-Union (IOU) measure as in [26]. As a complementary performance measure, we also compute the root mean squared error (RMSE) of the object orientation.

To assess the performance of the proposed method, we regard the Intersection-over-Union (IOU) and the root mean squared error (RMSE) of the object orientation. The definitions of these measures are given in the following: IOU$(\hat{S}, S)$ = area$(\hat{S} \cap S)$/area$(\hat{S} \cup S)$, RMSE$(\hat{\psi}) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(\hat{\psi}_k - \psi_k)^2}$, where the estimated and the true extent are represented by $\hat{S}$ and S, respectively; $\hat{\psi}$ and ψ indicate the estimated and the true orientation angle, respectively.

Time evolution of the kinematics $\bar{\mathbf{x}}_k$ is expressed by the nearly constant velocity model.

$$\bar{\mathbf{x}}_{k+1} = \bar{F}\bar{\mathbf{x}}_k + \bar{\mathbf{w}}_k, \quad \bar{\mathbf{w}}_k \sim \mathcal{N}(\mathbf{0}, \bar{Q}),$$

$$\bar{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes I_3, \; \bar{Q} = \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \otimes \begin{bmatrix} \sigma_c^2 & 0 & 0 \\ 0 & \sigma_c^2 & 0 \\ 0 & 0 & \sigma_\psi^2 \end{bmatrix},$$

where T is the sampling time; σ_c and σ_ψ indicate the process noise standard deviations for the center and the orientation, respectively; $\otimes$ denotes the Kronecker product. We describe the dynamics of the extent state by the following model, which provides the maximum entropy prediction density for unknown and slowly varying dynamics, [25, Theorem 1].

$$\mathbf{f}_{k+1} = \mathbf{f}_k + \mathbf{w}_k^f, \; \mathbf{w}_k^f \sim \mathcal{N}(\mathbf{0}, Q^f), \; Q^f = \left(\lambda^{-1} - 1\right) P_{k|k}^f, \tag{4.16}$$

where $P_{k|k}^f$ denotes the covariance of the estimated extent, and $\lambda < 1$ is a factor that scales up the estimated covariance to compute the predicted covariance, i.e, $P_{k+1|k}^f = \lambda^{-1} P_{k|k}^f$. Both of the methods employ the same GP model defined by the periodic covariance function, [26].

$$k(\theta, \theta') = \sigma_f^2 \exp(-\frac{2\sin^2\left(\frac{(\theta-\theta')}{2}\right)}{l^2}) + \sigma_r^2 \tag{4.17}$$

where σ_f^2 is the prior variance, l is the length-scale, and σ_r^2 is the constant term. The number of the variational iterations in VB is set to 10 for all experiments. Additionally, we implement another reference method, denoted as ‘CKF’, based on the cubature Kalman filter [83].

We investigate two distinct scenarios, which are indicated by S1 and S2. In S1 (model match scenario), the motion of the object is simulated by the nearly constant velocity model, and the trackers are provided with the true parameters of motion. In S2, the object traverses the path that is depicted in Fig. 4.2 at a constant speed v. In this setting, we examine a model mismatch between the true and the assumed motion dynamics.

Table 4.1: Mean Value of the Intersection-Over-Union (IOU)

		S1			**S2**		
		T1	**T2**	**T3**	**T1**	**T2**	**T3**
Exp1	**VB**	0.880	0.866	0.940	0.884	0.866	0.916
	EKF	0.875	0.863	0.937	0.882	0.869	0.913
	CKF	0.834	0.851	0.936	0.867	0.870	0.921
Exp2	**VB**	**0.838**	**0.777**	**0.929**	**0.829**	**0.784**	**0.881**
	EKF	0.569	0.557	0.827	0.691	0.650	0.805
	CKF	0.637	0.596	0.778	0.677	0.676	0.803
Exp3	**VB**	**0.814**	**0.783**	**0.918**	**0.827**	**0.766**	**0.869**
	EKF	0.655	0.631	0.830	0.696	0.644	0.795
	CKF	0.667	0.597	0.710	0.613	0.491	0.633

Table 4.2: RMSE of the Orientation Angle [rad]

		S1			**S2**		
		T1	**T2**	**T3**	**T1**	**T2**	**T3**
Exp1	**VB**	0.084	0.051	0.046	0.076	0.046	0.057
	EKF	0.094	0.052	0.046	0.082	0.044	0.060
	CKF	0.310	0.087	0.050	0.092	0.045	0.063
Exp2	**VB**	**0.314**	**0.443**	**0.090**	**0.183**	**0.224**	**0.110**
	EKF	1.200	1.060	0.399	0.850	0.534	0.389
	CKF	1.382	1.197	0.840	0.783	0.433	0.393
Exp3	**VB**	**0.409**	**0.387**	**0.082**	**0.209**	**0.325**	**0.212**
	EKF	0.997	0.844	0.438	0.913	0.569	0.401
	CKF	1.373	1.349	1.257	1.549	1.542	1.358

Table 4.3: Motion Variables Used in the Experiments with Simulated Data

	S1			**S2**			
	T (s)	σ_c (m)	σ_ψ (rad)	v (m/s)	T (s)	σ_c (m)	σ_ψ (rad)
Exp1	1	0.1	0.01	0.5	1	0.1	0.01
Exp2	1	1	0.1	2	1	0.5	0.05
Exp3	4	0.1	0.01	0.5	4	0.1	0.01

For both of the scenarios, we construct three separate experiments, denoted Exp1, Exp2 and Exp3, which present varying degrees of difficulty for a tracking application. More specifically, Exp1 corresponds to the least challenging environment by using a sufficiently small sampling time and involving modest levels of uncertainty in object's motion. In Exp2, the object of interest has greater maneuverability (e.g., using larger process noise for kinematics in S1, and increased speed in S2). Exp3 has the same motion characteristics with Exp1; however, the measurements are acquired less frequently as the sampling time is increased. Finally, for each experiment, we use three different extent models, i.e., rectangular (T1), triangular (T2) and a random extent sampled from a GP prior (T3).

Table 4.3 declares the values of the motion variables used in the experiments with simulated data. In particular, the given numbers specify both the simulation generation process and the parameters used by the trackers. As the first scenario (S1) examines a model match setting, the values of the motion variables, which are used while generating the ground truth, are identically provided to the trackers.

The hyperparameters of the GP model are set to $\sigma_f = 2$ m, $\sigma_r = 0.6$ m and $l = \pi/6$ rad; and the number of basis points is 50, which are evenly spaced in $[0, 2\pi]$. $\lambda = 0.99$ is used for the extent dynamics.

The mean of the IOU values and the RMSE of the object orientation are given in Tables 4.1 and 4.2, respectively. All numbers are averaged over 100 Monte Carlo runs with randomly generated measurements (for all experiments), trajectories (S1), and object extents (T3).

The results suggest that all methods are equally successful in Exp1. However, in Exp2 and Exp3, VB is significantly superior in regard to the estimation performance for the extent and the orientation. This observation is valid for both scenarios, S1 and S2, concerning the settings with model match and model mismatch. An example realization of the u-turn scenario with higher speed (S2-Exp2) using the rectangular extent model is illustrated in Fig. 4.2. Accordingly, we observe that VB establishes a more robust tracking algorithm at the cost of more computations. This can be attributed to the iterative nature of VB, which benefits multiple linearizations and expectation computations in a single measurement update.

4.3.2 Experiments with Real Data

In this section, the performance of the algorithms is demonstrated using real data. In particular, we investigate two experiments regarding different sensor modalities. Table 4.4 presents the corresponding results.

Experiment with Camera Data: In the first experiment, we consider information collected by a camera, mounted on an airborne vehicle which is steady over a surveillance region. The experiment takes place in a cove located at the western Mediterranean coast of Turkey. The scene involves a dinghy exhibiting agile maneuvers as shown in Fig. 4.3(a). Throughout the scenario, the camera captures an image in every 2 seconds. There are various algorithms that can extract features from camera images, such as Harris corner detector [84] and Lowe's Scale Invariant Feature Transform (SIFT) [85] (for an overview, see [86, Ch. 4]). To investigate a more general case, we consider point measurements generated by uniformly sampling the object's contour. The results obtained by using the features extracted by the Harris corner detector closely follow the results presented here, but they are not included in the manuscript because of the page limitations. The number of measurements on each frame is sampled from a Poisson distribution with average 10. Both methods share the identical set of parameters, which are specified as follows: The sampling time is set to $T = 2$ s; $R = 3^2 I_2$ m^2 is the measurement noise covariance matrix; the process noise standard deviations are used as $\sigma_c = 1$ m, $\sigma_\psi = 0.2$ rad; the same hyperparameters as defined in the simulation experiments are utilized for the GP model except $\sigma_f = 20$ m.

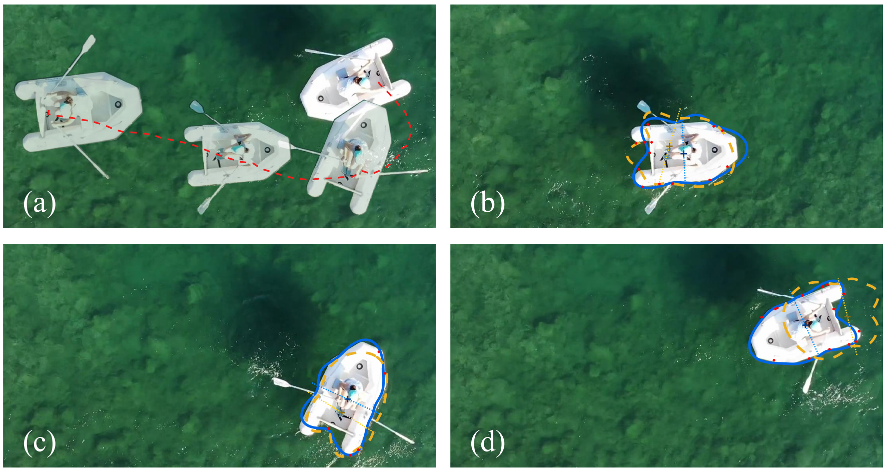


Figure 4.3: Tracking results for the video experiment. Fig. (a) shows the outline of the experiment. Starting from the leftmost position, the dinghy moves along the red dashed line. In Figs. (b)-(d), the estimates of the algorithms are plotted for three different frames (VB: —, EKF: - -, measurements: •).

Figs. 4.3(b)-(d) visualize the outputs of the algorithms for three instants. At the beginning of the experiment, while the dinghy moves along a relatively straight line, both methods perform successfully. However, once the vehicle starts maneuvering, VB exhibits favorable performance in terms of both the extent and the kinematic estimates as seen in Figs. (c) and (d).

Experiment with Laser Scanner Data: In the second experiment, we extract a sequence of point measurements acquired by a Velodyne HDL-64E laser scanner included in Kitti tracking benchmark, [72]. The outline of the scenario is illustrated by two example video frames in Fig. 4.4(b). The experiment takes place on an urban street where the ego and the target vehicles exhibit turn maneuvers while moving in the same direction.

Both methods use the same set of parameters, which are specified in the following: The sampling time is set to $T = 0.5$ s; $R = 0.8^2 I_2$ m^2 is the measurement noise covariance matrix; the process noise standard deviations are used as $\sigma_c = 3$ m, $\sigma_\psi = 0.8$ rad; the same hyperparameters as defined in the simulation experiments are utilized for the GP model except $l = \pi/5$ rad.

The results are demonstrated in Fig. 4.4(a). Both VB and EKF satisfactorily estimate

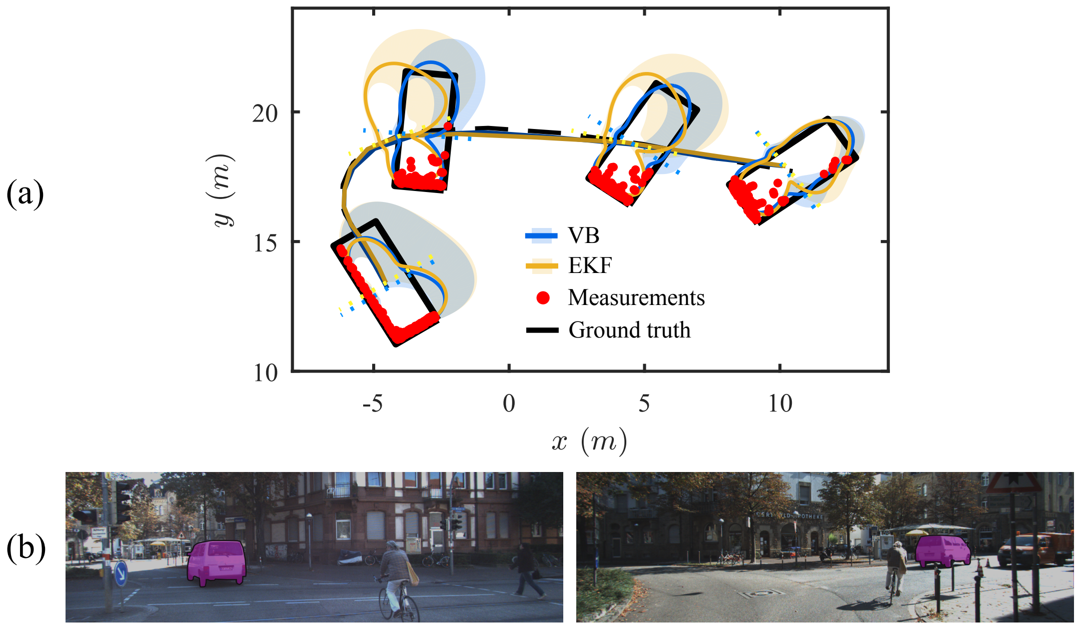


Figure 4.4: Fig. (a): Tracking results for the Kitti experiment (shaded areas represent 1-std confidence interval of the corresponding estimate). Fig. (b): Example video frames from the experiment.

the trajectory. Throughout the scenario, the sensor is mainly able to observe the left and rear parts of the target, thus the algorithms appropriately decrease the uncertainty on the observed portion of the extent while a larger uncertainty is associated with the unobserved section. However, during the maneuvering phase, VB outperforms EKF in orientation estimation.

Table 4.4: Results for the Experiments with Real Data

	Video Data		**Laser Scanner Data**	
	IOU	**RMSE($\hat{\psi}$) [rad]**	**IOU**	**RMSE($\hat{\psi}$) [rad]**
VB	**0.804**	**0.105**	**0.762**	**0.046**
EKF	0.672	0.363	0.624	0.276

4.4 Conclusion

We derived an alternative inference method for the GPEOT models using the variational Bayes techniques. The resulting algorithm is shown to significantly improve the tracking performance through experiments on simulated and real data (e.g., the percentage improvements in the IOU measure and in the accuracy of the orientation estimates exceed 40% and 400%, respectively). Additionally, it proves to be more robust for the scenarios with moderate or high levels of uncertainty included in the model. In this regard, we believe that the suggested method will enable the utilization of the GPEOT trackers in a wider range of real-world applications.

CHAPTER 5

EXTENDED OBJECT TRACKING USING REDUCED-RANK GAUSSIAN PROCESSES

5.1 Introduction

The literature on extended object tracking (EOT) is extensive (for detailed investigation of the existing studies, see [3, 80]). A great variety of trackers has been proposed relying on various mathematical representations of the unknown object extent. Seeking for a compact description of the extent, a large group of studies adopt simple parametric models, such as a stick [87, 88], a rectangle [5, 74], Multiple ellipses [89, 52] and B-splines [77, 90, 91] are also used to indicate the extent to attain more detailed representations.

For a flexible extent representation, another vein of studies assumes that the objects of interest are star-convex, and hence a radial distance function can be used to characterize the object shape. An important example of this class is the random hypersurface models, which parametrize the radial function by its Fourier series coefficients, [12, 92, 15, 93]. In a closely related line of literature, the unknown radial function is probabilistically modelled by a Gaussian process (GP) [26, 32].

The focus of this chapter is to improve the existing GP-based EOT (GPEOT) methods. GPEOT framework enables effective tracking of targets with unknown shapes using noisy point measurements, which can be collected by, for example, lidars, radars, and depth cameras. In particular, the resulting trackers jointly estimate the extent with the kinematic variables and are shown to perform successfully for various applications [29, 28, 53, 54, 27, 30, 34].

The key factor in the formulation of the GPEOT approach is the utilization of a GP model to describe the latent object extent. With their favorable analytical properties, GPs provide a convenient basis for the derivation of the mentioned methods. Additionally, they naturally account for the spatial correlation and the local uncertainty information in the extent estimates, which in turn facilitate effective shape learning using noisy and sparse observations. However, the computational complexity of obtaining the GP posterior imposes a great challenge for many practical settings [42]. Furthermore, the standard GP regression requires batch processing of all observations, which is not applicable for the typical tracking application as the number of available measurements sequentially grow over time. To alleviate these problems, the existing GPEOT framework employs an approximation for the GP model. The approximate approach maintains the radial function at finite number of inducing points, which are essentially the function values evaluated at uniformly distributed angles. Consequently, the resulting methods can recursively update the posterior of the inducing points together with the kinematics via standard inference techniques.

The mentioned approach requires to locate the inducing points at a sufficiently dense grid to precisely capture the characteristics of the underlying extent. The corresponding values of the radial function are included in the state vector, and their posterior is updated at each recursion of the tracker. Therefore, the memory storage and the computational load of the resulting method scale with the number of the inducing points. Using fewer points for an efficient implementation will reduce the representational power, which in turn degrades the tracking performance.

In this chapter, we derive GPEOT algorithms with improved computational characteristics without compromising the effective performance of the existing methods. To this end, we develop models based on a recent spectral-domain GP approximation, which complies well with the requirements of describing the unknown extent. In particular, we adopt the reduced-rank approximation [50] for the original GP model, which regards a basis function expansion of the Laplace operator for the stationary covariance function. The approximate GP model is essentially described by the weights of the basis functions. Consequently, the weights and the kinematic variables are recursively estimated using standard Bayesian techniques, which establishes an EOT approach. The contributions of this chapter can be summarized as follows.

- We propose a new formulation for the GPEOT models that facilitates high tracking performance at lower computational complexity.

- The improvement is enabled by the utilization of a recent reduced-rank GP approximation that offers an efficient description of the object extent. We perform necessary adaptations to apply the mentioned approximation in both two-dimensional (2D) and three-dimensional (3D) tracking problems. More specifically, we derive an approximate GP model for 2D targets that precisely implies periodic radial functions so that the estimated target extent is a closed shape without any discontinuities. Additionally, for the 3D trackers, we appropriately formulate the description of the 3D target extent that facilitates to specify an isotropic covariance function for the GP model as required by the reduced-rank approximation.

- The resulting extent representation is efficiently estimated together with the kinematic variables using standard inference techniques, such as an extended Kalman filter, which constitutes 2D and 3D GPEOT algorithms. Furthermore, we also investigate an implementation of the proposed trackers by relying on an iterated extended Kalman filter. The resulting method is comparatively evaluated with a recent study that uses variational iterations in a single measurement update under challenging tracking scenarios.

- The computational complexity of the suggested approach is analytically investigated in comparison with the existing methods.

- The performance of the resulting trackers is comprehensively demonstrated on simulated data, and a detailed analysis with regard to the key factors, e.g., number of basis functions, is presented.

5.2 Gaussian Processes

A Gaussian process (GP) is essentially a stochastic model that defines a probability distribution over a function space. Due to their flexibility in modeling, robustness to overfitting and principled uncertainty estimates, GPs have been widely adopted for various applications, [42]. In this chapter, we will utilize GPs to describe the latent extent of dynamic objects.

A GP model for the function $f(\cdot)$ is uniquely defined by

$$f(\mathbf{u}) \sim \mathcal{GP}\left(\mu(\mathbf{u}), k\left(\mathbf{u}, \mathbf{u}'\right)\right), \tag{5.1}$$

where $\mu(\cdot)$ is the mean function and $k(\cdot, \cdot)$ is the covariance function, which are specified as

$$\mu(\mathbf{u}) = \mathbb{E}[f(\mathbf{u})], \tag{5.2a}$$

$$k(\mathbf{u}, \mathbf{u}') = \mathbb{E}\left[(f(\mathbf{u}) - \mu(\mathbf{u}))(f(\mathbf{u}') - \mu(\mathbf{u}'))\right]. \tag{5.2b}$$

The GP indicates that any finite number of function values at arbitrary input points have a joint Gaussian distribution as in

$$\begin{bmatrix} f(\mathbf{u}_1) \\ \vdots \\ f(\mathbf{u}_M) \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, K), \tag{5.3}$$

where i^{th} entry of the mean vector $\boldsymbol{\mu}$ is defined as $\boldsymbol{\mu}_i = \mu(\mathbf{u}_i)$, and i-j^{th} entry of the covariance matrix K is defined as $K_{ij} = k(\mathbf{u}_i, \mathbf{u}_j)$.

5.2.1 Gaussian Process Regression

A GP provides a convenient basis to update the prior knowledge about the unknown function values in accordance with incomplete and noisy observations. In this context, let us consider the following model for the measurement y,

$$y = f(\mathbf{u}) + e, \quad e \sim \mathcal{N}\left(0, \sigma_r^2\right), \tag{5.4}$$

where e is the independent sensor noise, which has a Gaussian distribution with zero mean and variance σ_r^2.

The objective is to obtain a refined description of the function values $\mathbf{f}^* \triangleq \left[f(\mathbf{u}_1^*) \ \ldots \ f(\mathbf{u}_L^*)\right]^\top$ at some arbitrary inputs $\mathbf{u}^* \triangleq \left[\mathbf{u}_1^{*\top} \ \ldots \ \mathbf{u}_L^{*\top}\right]^\top$, by regarding the collected measurements $\mathbf{y} \triangleq \left[y_1 \ \ldots \ y_N\right]^\top$, which are noisy observations of the function evaluations at $\mathbf{u} \triangleq \left[\mathbf{u}_1^\top \ \ldots \ \mathbf{u}_N^\top\right]^\top$.

The proposed GP model together with the measurement model in (5.4) offers the following joint distribution

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(\mathbf{u}) \\ \boldsymbol{\mu}(\mathbf{u}^*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{u},\mathbf{u}) + I_N\sigma_r^2 & K(\mathbf{u},\mathbf{u}^*) \\ K(\mathbf{u}^*,\mathbf{u}) & K(\mathbf{u}^*,\mathbf{u}^*) \end{bmatrix} \right), \tag{5.5}$$

where I_N is the identity matrix of size N.

Consequently, by using the standard Gaussian conditioning formulae, we can analytically attain the posterior distribution of the function values as $p(\mathbf{f}^*|\mathbf{y}) \sim \mathcal{N}\left(\boldsymbol{\mu}^+, \Sigma^+\right)$, where the posterior mean and covariance are computed as

$$\begin{aligned} \boldsymbol{\mu}^+ &= \boldsymbol{\mu}(\mathbf{u}^*) + K\left(\mathbf{u}^*,\mathbf{u}\right)\left[K(\mathbf{u},\mathbf{u}) + I_N\sigma_r^2\right]^{-1}\left(\mathbf{y} - \boldsymbol{\mu}(\mathbf{u})\right) \\ \Sigma^+ &= K\left(\mathbf{u}^*,\mathbf{u}^*\right) - K\left(\mathbf{u}^*,\mathbf{u}\right)\left[K(\mathbf{u},\mathbf{u}) + I_N\sigma_r^2\right]^{-1}K\left(\mathbf{u},\mathbf{u}^*\right). \end{aligned} \tag{5.6}$$

5.2.2 Reduced-Rank Gaussian Processes

In a typical object tracking setting, the problem is to estimate the latent variables by processing measurements which become available in a sequential fashion. As the number of measurements monotonically increases over time, utilization of a naive GP model is infeasible due the ever-growing storage and computational requirements. Consequently, we hereby employ a spectral approximation of the GP model, which brings in favorable computational properties and an intuitive interpretation.

In [50], Solin and Särkkä present an approximation to GP modeling, which relies on the eigendecomposition of the negative Laplace operator on a confined domain. This approach leads to the following approximate expansion for isotropic covariance functions, i.e., $k(d) \triangleq k(\mathbf{u},\mathbf{u}')$, where $d \triangleq ||\mathbf{u} - \mathbf{u}'||$, and $||\cdot||$ denotes the Euclidean norm,

$$k_\Theta\left(\mathbf{u},\mathbf{u}'\right) \approx \sum_j S_\Theta(\sqrt{\lambda_j})\phi_j(\mathbf{u})\phi_j\left(\mathbf{u}'\right), \tag{5.7}$$

where $S_\Theta(\cdot)$ is the spectral density function, λ_j and $\phi_j(\cdot)$ are the j^{th} eigenvalue and eigenfunction of the negative Laplace operator solved in the specified domain, and Θ denotes the hyperparameters of the covariance function.

In this chapter, our interest relies in modeling the radial functions in compact domains. The resulting eigendecompositions will be computed analytically in closed-form as revealed in the subsequent sections. Note that if there is further prior information, one can choose the domain of interest accordingly and compute the corresponding decomposition. For example, in [94] the authors consider a two-dimensional hexagonal domain for the mapping application. Although a closed-form solution is not available, they effectively solve the eigendecomposition problem by numerical methods. Similarly, [95] investigates the reduced-rank approximation for GP models on domains with arbitrary boundaries, which include non-convex or disconnected spaces.

The central idea of the considered reduced-rank approximation is to truncate the expansion in (5.7) to a finite number of components,

$$k_{\Theta}\left(\mathbf{u}, \mathbf{u}'\right) = \sum_{j=1}^{m} S_{\Theta}(\sqrt{\lambda_j})\phi_j(\mathbf{u})\phi_j\left(\mathbf{u}'\right). \tag{5.8}$$

This is a reasonable approximation as the eigenvalues of the Laplace operator is monotonically increasing and the spectral density function goes to zero fast with increasing frequencies for bounded covariance functions, [50]. Interested readers can refer to [50] for detailed analysis and convergence proofs.

The truncated covariance function in (5.8) implies that the GP model can be represented by an equivalent linear model [42, Ch. 2.2] as in

$$f(\mathbf{u}) = \sum_{j=1}^{m} f_j\phi_j(\mathbf{u}), \text{ where } f_j \sim \mathcal{N}\left(0, S_{\Theta}(\sqrt{\lambda_j})\right). \tag{5.9}$$

This formulation suggests that the eigenfunctions of the Laplace operator $\{\phi_j(\cdot)\}_{j=1}^m$ form a set of basis functions to describe the underlying GP model, and the unknown weights $\{f_j\}_{j=1}^m$ have Gaussian prior distribution, whose variance is determined by the spectral density function. Note that one of the appealing properties of this decomposition is that the hyperparameters affect the expansion through the spectral density only, while the eigenfunctions depend exclusively on the choice of the domain.

We can reorganize (5.9) as

$$f(\mathbf{u}) = \boldsymbol{\varphi}(\mathbf{u})^{\top}\, \mathbf{f}, \text{ where } \mathbf{f} \sim \mathcal{N}\left(\mathbf{0}, S\right), \tag{5.10}$$

such that $\boldsymbol{\varphi}(\mathbf{u}) \triangleq [\phi_1(\mathbf{u}) \ldots \phi_m(\mathbf{u})]^\top$, $\mathbf{f} \triangleq [f_1 \ldots f_m]^\top$, and $S = \text{diag}\left(S_\Theta(\sqrt{\lambda_1}), \ldots, S_\Theta(\sqrt{\lambda_m})\right)$. This concludes the derivation of the reduced-rank approximation, where the original GP model is basically transformed into a linear-in-parameter model with the latent variable $\mathbf{f}$.

In accordance with (5.10), inferring the unknown radial function $f(\cdot)$ from a set of noisy observations $\{y_k\}_{k=1}^N$ amounts to computing the posterior distribution $p(\mathbf{f}|y_{1:N})$. As an example, let us define a generic observation model,

$$\begin{aligned} y_k &= f(\mathbf{u}) + e_k \\ &= \boldsymbol{\varphi}(\mathbf{u})^\top \mathbf{f} + e_k, \end{aligned} \tag{5.11}$$

where e_k stands for the independent sensor noise. Subsequently, we can apply Bayes' rule iteratively to expand the posterior as

$$\begin{aligned} p(\mathbf{f}|y_{1:N}) &\propto p(y_N|\mathbf{f}, y_{1:N-1})\, p(\mathbf{f}|y_{1:N-1}) \\ &= p(y_N|\mathbf{f})\, p(\mathbf{f}|y_{1:N-1}), \end{aligned} \tag{5.12}$$

where the second equation follows from the fact that $\mathbf{f}$ provides the sufficient statistics for y_k, i.e., $p(y_k|\mathbf{f}, y_{1:k-1}) = p(y_k|\mathbf{f})$ as implied by (5.11). Notice that this formulation will allow us to employ a standard Bayesian filter to recursively compute the posterior distribution. At each recursion of the filter, only the newly available measurements are processed to update the current estimate of the latent variable $\mathbf{f}$. This approach circumvents the computation and memory problems associated with the standard GP regression, which requires processing of all collected measurements in a single batch.

In the following sections, we will utilize this approximate GP model to construct state-space models regarding both the target extent and the kinematics, which in turn will be used to compute the joint posterior distribution in an efficient manner.

5.3 2D EOT Using Reduced-Rank Gaussian Processes

5.3.1 GP Modeling of 2D Object Extent

To be able to develop a probabilistic model of the object extent, we first need to establish a mathematical characterization of the extent. As mentioned earlier, there have been investigated many alternative representations in the literature. We hereby rely on one of the broadly adopted descriptions that expresses the target extent in polar coordinates by a radial function $f(\theta)$, [3, Section III-B]. In particular, the input of the radial function is the polar angle θ, and the output is the radial distance r between a reference point within the body and the object boundary, i.e., $r = f(\theta)$.

Notice that this representation implicitly assumes that the underlying object is star-convex[1]. This does not introduce a severe limitation in extent modeling and can be used to indicate a wide range of targets, see, for example, [92], [25].

Our objective is to track targets while simultaneously learning their unknown extent. To infer the target extent using noisy observations in a Bayesian framework, we need to provide a probabilistic model for the extent. Introduced in [26], the key idea behind the GPEOT approach is to model the latent extent by a GP. More specifically, a GP prior is imposed on the latent radial function as

$$f(\theta) \sim \mathcal{GP}\left(0, k(\theta, \theta')\right), \tag{5.13}$$

where the mean function is assumed to be identically zero for brevity, and the covariance function, $k(\theta, \theta')$, basically defines the correlation between the function evaluations at the polar angles, θ and θ'. This formulation naturally accounts for the inherent spatial correlation within the object body and provides a systematic update mechanism to accumulate the shape information.

The aforementioned practical problems of the standard GP regression requires us to approximate the original model for utilizing the sequentially available measurements in a object tracking application. In this chapter, we employ the reduced-rank approximation presented in Section 5.2.2 for the GP model. The method is based on the series expansion of the covariance function in terms of the eigendecomposition of the negative Laplace operator. Consequently, for the univariate case, the problem is to identify the eigenvalue and eigenfunction pairs in a compact subset $\Omega = [-L, L] \subset \mathbb{R}$ using some boundary condition, such as the Dirichlet boundary condition, i.e.,

$$\begin{aligned} -\nabla^2 \phi_j(\theta) &= \lambda_j^2 \phi_j(\theta), \quad && \theta \in \Omega, \\ \phi_j(\theta) &= 0, && \theta \in \partial\Omega, \end{aligned} \tag{5.14}$$

[1] A set $\mathcal{S}$ is called star-convex if there exist a point ζ_0 such that any line segment from ζ_0 to any other point in $\mathcal{S}$ is fully contained in S.

where ∇^2 denotes the Laplace operator.

Notice that the fidelity of the approximated model might degrade near the boundary of Ω as the Dirichlet condition drives the process through zero, [50]. This is considered to be a minor problem for many applications, as one can simply avoid these undesired boundary effects by extending the specified domain by a factor and performing inference within an inner section of the domain which leaves some margin from the boundaries, [50, 94].

In our formulation, the estimated radial function is strictly required to be periodic with 2π, i.e., $f(\theta) = f(\theta + 2\pi)$, to imply a closed target shape. The periodicity of the radial function is encoded by appropriately specifying a periodic covariance function for the GP model [26, Section IV-B]. In this regard, the above-mentioned workaround for the boundary effects potentially eliminates the periodicity of the approximated covariance function in the domain of interest, which in turn might lead to a discontinuity in the estimated object extent. As a solution, we adapt the eigenvalue problem for modelling the radial function as

$$\begin{aligned} -\nabla^2 \phi_j(\theta) &= \lambda_j^2 \phi_j(\theta), \quad && \theta \in \Omega, \\ \phi_j(\theta) &= c_j, && \theta \in \partial\Omega, \end{aligned} \tag{5.15}$$

where the domain is set to $\Omega = [0, 2\pi]$. This formulation assures that the resulting eigenfunctions evaluate the same value c_j at the domain boundary, i.e., 0 and 2π, thus a possible discontinuity in the estimated extent is avoided.

The corresponding eigenvalues and eigenfunctions are analytically obtained as

$$\begin{aligned} \phi_j^{(a)}(\theta) &= \cos\left(\frac{2j\pi\theta}{L}\right), \quad \phi_j^{(b)}(\theta) = \sin\left(\frac{2j\pi\theta}{L}\right), \\ \lambda_j^{(a)} &= \lambda_j^{(b)} = \frac{2\pi j}{L}, \end{aligned} \tag{5.16}$$

where $L = 2\pi$. With this decomposition, the corresponding approximate covariance function is now guaranteed to be periodic with 2π as desired. The analytic expression for the approximate covariance function is given as

$$k\left(\theta, \theta'\right) = \sum_{j=1}^{m} S\left(\sqrt{\frac{2\pi j}{L}}\right) \cos\left(\frac{2j\pi(\theta - \theta')}{L}\right). \tag{5.17}$$

Consequently, we finalize the reduced-rank GP model of the object extent as

$$f(\theta) = \boldsymbol{\varphi}(\theta)^\top \mathbf{f}, \text{ where } \mathbf{f} \sim \mathcal{N}(\mathbf{0}, S), \tag{5.18a}$$

$$\boldsymbol{\varphi}(\theta) = \left[\phi_1^{(a)}(\theta),\, \phi_1^{(b)}(\theta),\, \ldots,\, \phi_m^{(a)}(\theta),\, \phi_m^{(b)}(\theta)\right]^\top, \tag{5.18b}$$

$$\mathbf{f} = \left[f_1^{(a)},\, f_1^{(b)},\, \ldots,\, f_m^{(a)},\, f_m^{(b)}\right]^\top, \tag{5.18c}$$

$$S = \text{diag}\left(S(\sqrt{\lambda_1^{(a)}}), S(\sqrt{\lambda_1^{(b)}}), \ldots, S(\sqrt{\lambda_m^{(a)}}), S(\sqrt{\lambda_m^{(b)}})\right), \tag{5.18d}$$

where $\mathbf{f}$ denotes the parametrized description of the extent.

5.3.2 State-Space Model

In this section, we will develop a unified state-space model regarding both the kinematics and the extent of the target, which in turn will enable the joint estimation of the variables.

Consider the augmented state vector at time k

$$\mathbf{x}_k \triangleq \begin{bmatrix}\bar{\mathbf{x}}_k^\top & \mathbf{f}_k^\top\end{bmatrix}^\top, \tag{5.19}$$

where $\mathbf{f}_k$ indicates the parametrized description of the target extent, and $\bar{\mathbf{x}}_k$ includes the kinematic state variables, defined as

$$\bar{\mathbf{x}}_k \triangleq \begin{bmatrix}\mathbf{c}_k^\top & \psi_k & \mathbf{v}_k^\top & \omega_k\end{bmatrix}^\top. \tag{5.20}$$

where $\mathbf{c}_k$ is the 2D position of a reference point within the object body, which can be interpreted to be the object center, ψ_k is the target orientation, $\mathbf{v}_k$ is the linear velocity of the center, and ω_k is the angular rate.

5.3.2.1 Measurement Model

We will first develop a measurement model that expresses the relation between the sensor observations and the variables included in the state vector. To this end, we define two different coordinate frames. First, the global coordinate frame, $\{G\}$, is fixed to the sensor, in which the measurements are resolved. Second, the local coordinate frame, $\{L\}$, is anchored to the tracked object. These coordinate frames are illustrated in Fig. 5.1 using an example extended target.

Let us consider a single point measurement $\mathbf{y}_{k,l} \in \mathbb{R}^2$ collected at time k. By the superposition of the vectors, we can write

$$\mathbf{y}_{k,l} = \mathbf{c}_k + \mathbf{p}_{k,l}\, f(\theta_{k,l}) + \mathbf{e}_{k,l},\ \mathbf{e}_{k,l} \sim \mathcal{N}(\mathbf{0}, R), \tag{5.21}$$

where $\mathbf{c}_k$ is the object center, $\mathbf{p}_{k,l}$ is a unit-length vector along the direction from the object center to the measurement source, $f(\cdot)$ is the radial function, $\theta_{k,l}$ is the polar angle corresponding to the measurement source, and $\mathbf{e}_{k,l}$ is the independent Gaussian measurement noise, with zero mean and covariance R.

Note that the source point that originates the measurement $\mathbf{y}_{k,l}$ is not available to the sensor in practice. Therefore, we need to substitute some approximate expressions for the variables $\mathbf{p}_{k,l}$ and $\theta_{k,l}$, which will be regarded in the derivation of the measurement model. In this respect, we define $\mathbf{p}_{k,l}$ as

$$\mathbf{p}_{k,l} = \mathbf{p}_l\left(\mathbf{c}_k\right) = \frac{\mathbf{y}_{k,l} - \mathbf{c}_k}{\|\mathbf{y}_{k,l} - \mathbf{c}_k\|}. \tag{5.22}$$

Next, we can compute the polar angle associated with the vector from the object center to the measurement in the global coordinate as

$$\theta^G_{k,l}\left(\mathbf{c}_k\right) = \angle\left(\mathbf{y}_{k,l} - \mathbf{c}_k\right), \tag{5.23}$$

which is then transformed into the local coordinate frame by

$$\theta^L_{k,l}\left(\mathbf{c}_k, \psi_k\right) = \theta^G_{k,l}\left(\mathbf{c}_k\right) - \psi_k. \tag{5.24}$$

By substituting these approximations into the measurement equation, we obtain

$$\mathbf{y}_{k,l} = \mathbf{c}_k + \mathbf{p}_l\left(\mathbf{c}_k\right) f\left(\theta^L_{k,l}\left(\mathbf{c}_k, \psi_k\right)\right) + \mathbf{e}_{k,l}. \tag{5.25}$$

Finally, by plugging the reduced-rank GP description for the radial function, we end up with the following measurement model.

$$\begin{aligned} \mathbf{y}_{k,l} &= \underbrace{\mathbf{c}_k + \mathbf{p}_l\left(\mathbf{c}_k\right)\ \boldsymbol{\varphi}\left(\theta^L_{k,l}\left(\mathbf{c}_k, \psi_k\right)\right)^\top\ \mathbf{f}_k}_{=\mathbf{h}_l(\mathbf{x}_k)} + \mathbf{e}_{k,l} \\ &= \mathbf{h}_l\left(\mathbf{x}_k\right) + \mathbf{e}_{k,l} \end{aligned} \tag{5.26}$$

An important distinction of the final measurement model from the one derived in [26] is that the statistics of the sensor noise in (5.26) do not depend on the unknown state variables, instead they are exclusively determined by the sensor characteristics. In this respect, the standard techniques of nonlinear Kalman filtering are directly applicable for inference without further assumptions.

5.3.2.2 Process Model

To complete the construction of the state-space model, we will define a mathematical model that characterizes the time evolution of the state vector.

The formulation of our approach does not impose any restriction on the selection of the process model. We utilize the following linear Gaussian model,

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{w}_k, \ \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q_k), \tag{5.27}$$

where F is the system matrix, and $\mathbf{w}_k$ denotes the process noise, which is Gaussian with zero mean and covariance Q_k.

Assuming that the kinematic and the extent variables do not interact with each other in time, we can specify the overall process model as a composition of two independent subblocks

$$F = \begin{bmatrix} \bar{F} & 0 \\ 0 & F^{\mathbf{f}} \end{bmatrix}, \ Q_k = \begin{bmatrix} \bar{Q} & 0 \\ 0 & Q_k^{\mathbf{f}} \end{bmatrix}. \tag{5.28}$$

For the kinematics state, we utilize a nearly constant velocity model,

$$\bar{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes I_3, \ \bar{Q} = \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \otimes \begin{bmatrix} \sigma_c^2 & 0 & 0 \\ 0 & \sigma_c^2 & 0 \\ 0 & 0 & \sigma_\psi^2 \end{bmatrix}, \tag{5.29}$$

where T denotes the sampling time, σ_c and σ_ψ denote the process noise standard deviations for the center and the orientation, respectively; $\otimes$ is the Kronecker product.

We define the dynamics of the extent state by

$$F^{\mathbf{f}} = I, \quad Q_k^{\mathbf{f}} = \left(\lambda^{-1} - 1\right) P_{k|k}^{\mathbf{f}}. \tag{5.30}$$

$P_{k|k}^{\mathbf{f}}$ is the covariance of the estimated extent state at time k. This model is shown to provide the maximum entropy prediction density for unknown and slowly varying dynamics [59, Theorem 1] which is assumed to comply with the characteristics of the object extent. Notice that the model leads to a prediction density, whose mean is the same with that of the estimated density and covariance is effectively scaled up as $P_{k+1|k}^{\mathbf{f}} = \lambda^{-1} P_{k|k}^{\mathbf{f}}$ for $\lambda < 1$.

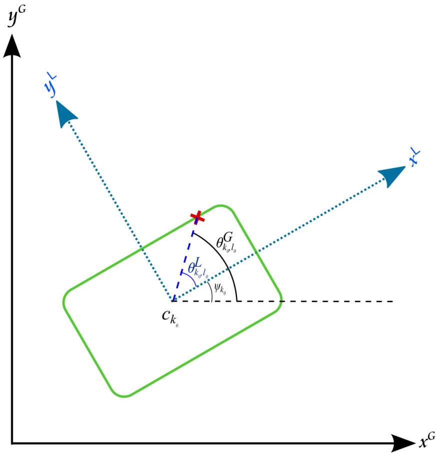


Figure 5.1: Illustration of the state variables regarded in the derivation of the 2D EOT algorithm.

Furthermore, the prior distribution of the state vector is specified as

$$\mathbf{x}_0 \sim \mathcal{N}\left(\boldsymbol{\mu}_0, P_0\right) \tag{5.31a}$$

$$\text{where } \boldsymbol{\mu}_0 = \begin{bmatrix} \bar{\boldsymbol{\mu}}_0 \\ \boldsymbol{\mu}_0^{\mathbf{f}} \end{bmatrix}, \; P_0 = \begin{bmatrix} \bar{P}_0 & 0 \\ 0 & P_0^{\mathbf{f}} \end{bmatrix}, \tag{5.31b}$$

which suggests that the kinematic and the extent variables have independent Gaussian prior distributions, i.e., $\bar{\mathbf{x}}_0 \sim \mathcal{N}\left(\bar{\boldsymbol{\mu}}_0, \bar{P}_0\right)$ and $\mathbf{f}_0 \sim \mathcal{N}\left(\boldsymbol{\mu}_0^{\mathbf{f}}, P_0^{\mathbf{f}}\right)$. Note that the prior distribution for the extent variables is directly offered by the reduced-rank GP model as detailed in (5.18).

This concludes the derivation of our state-space model for 2D extended targets. We will regard this model to infer the latent kinematic and the extent variables in a standard Bayesian framework using point measurements. The details of the utilized inference scheme will be presented in Section 5.5.

5.4 3D EOT Using Reduced-Rank Gaussian Processes

In this section, we will introduce a 3D object tracking approach that can jointly estimate the kinematics and the shape of a 3D object by processing 3D point measurements. To this end, we develop an efficient GP model for the latent 3D extent based on the reduced-rank approximation and construct a state-space model regarding both the kinematics and the extent description.

5.4.1 GP Modeling of 3D Target Extent

We first need to formulate a mathematical description for the 3D shape, which in turn will be considered to develop a probabilistic model. To this end, we exclusively regard the external boundary (surface) of the object. Additionally, assuming that the underlying object is star-convex, we can express this 3D surface by a radial function.

In the literature, the radial function representation is shown to lead to effective solutions for the 3D tracking problem, e.g., [93, 32, 34, 96]. In particular, the mentioned methods describe the target shape by a radial function $r = f(\theta, \phi)$ in spherical coordinates, where the output r is the radial distance between the target center and the corresponding point on the surface implied by the spherical angle pair, i.e., the azimuth, $\theta \in [-\pi, \pi]$, and the elevation angles, $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Subsequently, it is possible to model the unknown radial function by a GP. In this pursuit, the main challenge is to construct a proper covariance function for the GP, which conforms to the principles of the spherical geometry. More specifically, there is a need for specifying an appropriate distance measure to be used in the covariance function, which calculates the relative distance between two spherical coordinates. For example, the standard choice of the Euclidean distance will lead to an erroneous correlation structure for the extent described in the spherical coordinates [34]. In this regard, in [32, 34] the distance is considered to be the angle of the shortest arc on a sphere, or the geodesic distance, which enables to induce the correct correlation patterns in the spherical coordinates.

As indicated previously, the reduced-rank approximation requires the original GP to use an isotropic covariance function. Unfortunately, the mentioned geodesic distance does not comply with this requirement, hence it is not possible to directly apply the reduced-rank approximation to the given GP model. As a solution to this problem, we will hereby formulate a different description for the target shape, where the argument of the radial function is chosen to be a point on the unit sphere, i.e.,

$$f(\mathbf{p}) = r \tag{5.32}$$

where $\mathbf{p} \in \mathbb{S}^2$, and $\mathbb{S}^2 := \{\mathbf{p} \in \mathbb{R}^3 \mid \|\mathbf{p}\| = 1\}$. Consequently, the corresponding GP

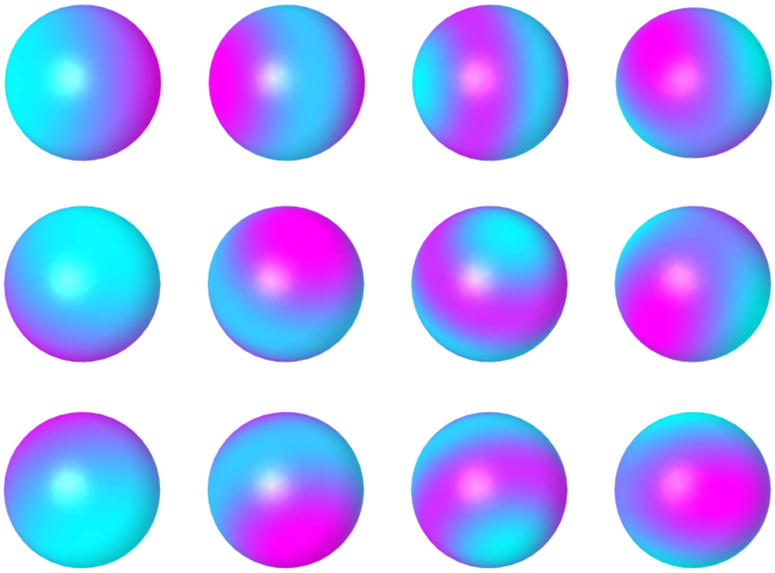


Figure 5.2: Representative eigenfunctions visualized over a spherical surface. The largest and the smallest values of the functions are mapped to solid magenta and cyan, respectively; the values in between are plotted according to a linear interpolation. (Light reflections are added to the illustrations to facilitate 3D interpretation.)

model can be indicated as

$$f(\mathbf{p}) \sim \mathcal{GP}\left(0, k\left(\mathbf{p}, \mathbf{p}'\right)\right). \tag{5.33}$$

This representation will enable us to define an isotropic covariance function for the GP model, for example, by using the Euclidean distance, i.e., $k(d) = k(\mathbf{p}, \mathbf{p}')$ where $d = ||\mathbf{p} - \mathbf{p}'||$. In this respect, we can easily adopt the widely utilized covariance functions, such as the squared exponential or a member of the Matérn class [42].

Thereafter, to obtain the reduced-rank approximation of the given GP model, we solve the eigendecomposition of the negative Laplace operator, $-\nabla^2$, subject to the Dirichlet boundary condition on a confined domain Ω, as in the following.

$$\begin{aligned} -\nabla^2\phi_j(\mathbf{p}) &= \lambda_j^2\phi_j(\mathbf{p}), \quad && \mathbf{p} \in \Omega \\ \phi_j(\mathbf{p}) &= 0, && \mathbf{p} \in \partial\Omega \end{aligned} \tag{5.34}$$

Without imposing any prior assumption, we choose the domain of interest to be a rectangular prism, i.e., $\Omega = [-L_1, L_1] \times [-L_2, L_2] \times [-L_3, L_3]$, for which the eigenvalues and the eigenfunctions are analytically computed as

$$\phi_{j_1,j_2,j_3}(\mathbf{p}) = \prod_{d=1}^{3} \frac{1}{\sqrt{L_d}} \sin\left(\frac{\pi j_d\,(p_d + L_d)}{2L_d}\right), \tag{5.35a}$$

$$\lambda_{j_1,j_2,j_3} = \sum_{d=1}^{3} \frac{\pi j_d}{2L_d}, \tag{5.35b}$$

where p_d is the d^{th} component of the input vector $\mathbf{p}$, and (j_1, j_2, j_3) is an index tuple. Some of the eigenfunctions illustrated over a spherical surface is depicted in Fig. 5.2.

Finally, the resulting approximate model of the radial function can be stated as

$$f(\mathbf{p}) = \boldsymbol{\varphi}(\mathbf{p})^\top \mathbf{f}, \text{ where } \mathbf{f} \sim \mathcal{N}(\mathbf{0}, S), \tag{5.36}$$

where $\boldsymbol{\varphi}(\cdot)$ is formed by concatenating the eigenfunctions, and $\mathbf{f}$ consists of the corresponding coefficients.

5.4.2 State-Space Model

In this section, we develop a state-space model that determines the dynamical characteristics of the latent variables and how they relate to the observations. To enable joint estimation of the unknown quantities, the state vector consists of both the kinematics $\bar{\mathbf{x}}_k$ and the extent variables $\mathbf{f}_k$, i.e., $\mathbf{x}_k \triangleq \left[\bar{\mathbf{x}}_k^\top \; \mathbf{f}_k^\top\right]^\top$.

5.4.2.1 Measurement Model

We first derive the measurement model which reveals the relation between the 3D point measurements and the state variables. Similarly to the 2D setting, we make use of two distinct coordinate frames, namely the global and the local coordinate frame, which are fixed to the sensor and the target, respectively.

A single 3D point measurement collected at time k $\mathbf{y}_{k,l} \in \mathbb{R}^3$ can be expressed as

$$\mathbf{y}_{k,l} = \mathbf{c}_k + \mathbf{p}_{k,l}\, f(\mathbf{p}_{k,l}^L) + \mathbf{e}_{k,l},\; \mathbf{e}_{k,l} \sim \mathcal{N}(\mathbf{0}, R). \tag{5.37}$$

$\mathbf{c}_k$ is the target center, $f(\cdot)$ is the radial function, $\mathbf{p}_{k,l}^L$ indicates the point on the unit sphere corresponding to the measurement source on the target surface originating

$\mathbf{y}_{k,l}$, and it is expressed in the local coordinate frame, $\mathbf{p}_{k,l}$ is the unit-length vector pointing from the target center to the the measurement source, and $\mathbf{e}_{k,l}$ stands for the independent zero-mean Gaussian measurement noise with covariance R.

As the true source point that gives rise to the measurement $\mathbf{y}_{k,l}$ is unknown, we approximate the corresponding $\mathbf{p}_{k,l}$ and $\mathbf{p}^L_{k,l}$ in (5.37) by directly referring to measurement instead of the source point as in the following.

$$\mathbf{p}_{k,l} = \mathbf{p}_l\left(\mathbf{c}_k\right) = \frac{\mathbf{y}_{k,l} - \mathbf{c}_k}{\|\mathbf{y}_{k,l} - \mathbf{c}_k\|} \tag{5.38}$$

$$\begin{aligned} \mathbf{p}^L_{k,l} = \mathbf{p}^L_l\left(\mathbf{c}_k, \mathbf{q}_k\right) &= \frac{\mathbf{y}^L_{k,l}\left(\mathbf{c}_k, \mathbf{q}_k\right)}{\|\mathbf{y}^L_{k,l}\left(\mathbf{c}_k, \mathbf{q}_k\right)\|} \\ \text{where } \mathbf{y}^L_{k,l}\left(\mathbf{c}_k, \mathbf{q}_k\right) &= R^L_G(\mathbf{q}_k)\left(\mathbf{y}_{k,l} - \mathbf{c}_k\right) \end{aligned} \tag{5.39}$$

$R^L_G(\cdot)$ is the rotation matrix from the global to the local frame, and $\mathbf{q}_k$ is the unit quaternion that declares the target orientation. Note that $\mathbf{p}^L_{k,l}$, which will be fed to the radial function as the argument, is located on the unit sphere as the definition of the function requires.

Then, we can rewrite the measurement equation in (5.37) as

$$\mathbf{y}_{k,l} = \mathbf{c}_k + \mathbf{p}_l(\mathbf{c}_k) f\left(\mathbf{p}^L_l\left(\mathbf{c}_k, \mathbf{q}_k\right)\right) + \mathbf{e}_{k,l}\ . \tag{5.40}$$

Finally, we substitute the reduced-rank GP description for the radial function given in (5.36) and obtain the measurement model as

$$\begin{aligned} \mathbf{y}_{k,l} &= \underbrace{\mathbf{c}_k + \mathbf{p}_l\left(\mathbf{c}_k\right)\ \boldsymbol{\varphi}\left(\mathbf{p}^L_l\left(\mathbf{c}_k, \mathbf{q}_k\right)\right)^\top\ \mathbf{f}_k}_{=\mathbf{h}_l(\mathbf{x}_k)} + \mathbf{e}_{k,l} \\ &= \mathbf{h}_l\left(\mathbf{x}_k\right) + \mathbf{e}_{k,l}\ . \end{aligned} \tag{5.41}$$

5.4.2.2 Process Model

We reveal the process model of the state vector to be regarded within the state-space model in this section. Note that the formulation of the proposed tracker does not put any limitation on the specification of the process model. We hereby essentially adopt the model introduced in [34], which can be summarized as a linear Gaussian model

$$\mathbf{x}_{k+1} = F_k \mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q_k)\ , \tag{5.42}$$

where F_k is the system matrix, and $\mathbf{w}_k$ indicates the process noise, which is Gaussian with zero mean and covariance Q_k.

Remember that the state comprises of the kinematics and the extent, $\mathbf{x}_k = \left[\bar{\mathbf{x}}_k^\top \; \mathbf{f}_k^\top\right]^\top$. We assume that the dynamics of these two state components do not interact with each other in time. Accordingly, the process model can be written to include two independent subsystems as

$$F_k = \begin{bmatrix} \bar{F}_k & 0 \\ 0 & F^{\mathbf{f}} \end{bmatrix}, \; Q_k = \begin{bmatrix} \bar{Q}_k & 0 \\ 0 & Q_k^{\mathbf{f}} \end{bmatrix}. \tag{5.43}$$

For the extent state, we use the same dynamical model with the one used in the 2D tracker,

$$F^{\mathbf{f}} = I, \quad Q_k^{\mathbf{f}} = \left(\lambda^{-1} - 1\right) P_{k|k}^{\mathbf{f}} \, . \tag{5.44}$$

$P_{k|k}^{\mathbf{f}}$ is the covariance of the estimated extent state at time k.

The kinematics state includes two parts describing the translational and the rotational state, denoted by $\mathbf{x}_k^t$ and $\mathbf{x}_k^r$, respectively, i.e., $\bar{\mathbf{x}}_k \triangleq \left[\mathbf{x}_k^{t\top} \; \mathbf{x}_k^{r\top}\right]^\top$. By assuming the independence of these elements, we write

$$\bar{F}_k = \begin{bmatrix} F^t & 0 \\ 0 & F_k^r \end{bmatrix}, \; \bar{Q}_k = \begin{bmatrix} Q^t & 0 \\ 0 & Q_k^r \end{bmatrix}. \tag{5.45}$$

We consider the target center $\mathbf{c}_k \in \mathbb{R}^3$ and the velocity $\mathbf{v}_k \in \mathbb{R}^3$ to describe the translational kinematics, i.e., $\mathbf{x}_k^t \triangleq \left[\mathbf{c}_k^\top \; \mathbf{v}_k^\top\right]^\top$. Furthermore, the translational motion is modelled by the nearly constant velocity model,

$$F^t = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes I_3, \; Q^t = \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \otimes \left(\sigma_c^2 I_3\right) , \tag{5.46}$$

where σ_c is the process noise standard deviation for the center.

For the rotational motion, we employ the constant velocity model derived in [34]. The model is based on an effective description of the 3D target orientation, which is free of singularities and lends itself conveniently to Bayesian estimation. In particular, the utilized representation makes use of a reference orientation as a unit quaternion,

$\mathbf{q} \in \mathbb{R}^4$, $\|\mathbf{q}\| = 1$, and the orientation deviation from this reference, which is expressed by the vector, $\mathbf{a} \in \mathbb{R}^3$. The objective is to estimate the deviation vector $\mathbf{a}$ in a Bayesian framework, which in turn periodically updates the reference orientation accordingly. The resulting approach is known as the multiplicative extended Kalman filter (MEKF) in the literature, [61]. The rotational kinematics vector is defined to include the mentioned deviation vector $\mathbf{a}$ and the angular velocity of the local frame with respect to the global frame $\boldsymbol{\omega} \triangleq [\omega_x \; \omega_y \; \omega_z]^\top$, i.e., $\mathbf{x}^r \triangleq \left[\mathbf{a}^\top \; \boldsymbol{\omega}^\top\right]^\top$. Full details of the rotational motion model are revealed in Chapter 3.5.1.2 and Appendix A.4 and will not be repeated in this section.

5.5 Inference

In the previous sections, relying on an efficient approximation for the extent description, we derived state-space models for both 2D and 3D objects. The final step for constructing the corresponding EOT algorithms is to adopt a mechanism to infer the latent state vectors using collected observations. Treating the problem in a Bayesian framework, the objective is to compute the posterior distribution of the state vector.

$\{\mathbf{y}_{k,l}\}_{l=1}^{n_k}$ denotes the multiple measurements acquired at time k. To instantaneously process all of the measurements, it is possible to concatenate them and express the corresponding measurement model as

$$\mathbf{y}_k = \left[\mathbf{y}_{k,1}^\top, \; \ldots, \; \mathbf{y}_{k,n_k}^\top\right]^\top, \tag{5.47a}$$

$$\mathbf{h}\left(\mathbf{x}_k\right) = \left[\mathbf{h}_1\left(\mathbf{x}_k\right)^\top, \; \ldots, \; \mathbf{h}_{n_k}\left(\mathbf{x}_k\right)^\top\right]^\top, \tag{5.47b}$$

$$R_k = I_{n_k} \otimes R. \tag{5.47c}$$

Subsequently, we can immediately rewrite the resulting state-space model

$$\begin{aligned} \mathbf{x}_{k+1} &= F\mathbf{x}_k + \mathbf{w}_k, \; \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q) \\ \mathbf{y}_k &= \mathbf{h}\left(\mathbf{x}_k\right) + \mathbf{e}_k, \; \mathbf{e}_{k,l} \sim \mathcal{N}(\mathbf{0}, R_k) \\ \mathbf{x}_0 &\sim \mathcal{N}\left(\boldsymbol{\mu}_0, P_0\right). \end{aligned} \tag{5.48}$$

There is a wide spectrum of standard Bayesian methods that might be utilized to compute the posterior distribution of the state vector regarding the model in (5.48).

In this study, we will essentially employ an extended Kalman filter (EKF) due to the nonlinearity in the measurement model. The details of the inference mechanism are given in Appendix A.2.1. This concludes the derivation of the proposed EOT approach, which efficiently updates the posterior of the joint state vector at each time instant by processing the newly available measurements.

5.6 Simulations and Results

In this section, we will demonstrate the performance of the proposed formulation for the GPEOT algorithms introduced in this chapter. Throughout this section, we denote the suggested approach for the GPEOT framework as ‘FGP’ (short for fast GPEOT) while the standard approach is referred to as ‘IGP’ (considering the model maintains the extent at the inducing inputs.)

5.6.1 Analytical Analysis of the Computational Complexity

In this section, we will analytically analyze the computational advantages of the proposed formulation for the GPEOT algorithms over its existing counterpart [26]. We will specifically focus on the disparity in the number of operations required to perform in a single recursion of the resulting tracking filters.

As mentioned earlier, both of the FGP and IGP approaches develop state-space models that regard kinematic variables and a GP representation of the latent target extent. In this pursuit, they apply distinct approximations to attain a parametric description of the underlying GP model. Consequently, to establish the corresponding EOT algorithms, a standard Bayesian inference technique, which utilizes the mentioned state-space model, is adopted. The resulting filters recursively compute the posterior distribution of the state vector, which comprises of both the kinematic and the extent variables. In this analysis, we assume that both of the methods employ the same inference technique to ensure a fair comparison.

The standard Bayesian recursion consists of the following consecutive procedures: time prediction and measurement update. We assume that the methods utilize the same process model to express the time evolution of the state vector. Therefore, the computational complexity of obtaining the predicted distribution through the time prediction is identical for the methods. However, the computational characteristics of the methods differentiate during the measurement update. In this phase, the predicted distribution is corrected by the employed inference mechanism in accordance with the observations, regarding the measurement predictions and the associated covariances. The computational discrepancy arises due to the different formulation of the measurement models, which rely on alternative derivations of the approximate GP model for the unknown radial function.

Let us assume that at time k, the algorithms process the measurements $\{\mathbf{y}_{k,l}\}_{l=1}^{n}$ to compute the posterior distribution, where n is the number of available measurements. The measurement models of FGP and IGP are revealed in Section 5.3.2.1 and Appendix A.6, respectively. Note that we explicitly refer to the formulation of the 2D trackers, as the computational analysis is identical for the 3D setting.

The discrepancy is caused by the formulation of the approximate GP models of the latent radial function. In particular, for each measurement $\mathbf{y}_{k,l}$, IGP requires to calculate the matrix $H^{\mathbf{f}}_{k,l}$ in (A.14) given by

$$H^{\mathbf{f}}_{k,l} = k(\theta^{L}_{k,l}, \boldsymbol{\theta}^{\mathbf{f}})\ K(\boldsymbol{\theta}^{\mathbf{f}}, \boldsymbol{\theta}^{\mathbf{f}})^{-1} \ ,$$

where $\theta^{L}_{k,l}$ is the input angle associated with the measurement, $\boldsymbol{\theta}^{\mathbf{f}} \triangleq \left[\theta^{\mathbf{f}}_{1} \dots \theta^{\mathbf{f}}_{m}\right]^{\top}$ denotes the vector including the inducing input angles, and $K(\boldsymbol{\theta}^{\mathbf{f}}, \boldsymbol{\theta}^{\mathbf{f}}) \in \mathbb{R}^{m \times m}$ is the covariance matrix of the function values evaluated at the input angles. For a standard setting, the inducing inputs are fixed during the operation of the filter, hence the inverse of $K(\boldsymbol{\theta}^{\mathbf{f}}, \boldsymbol{\theta}^{\mathbf{f}})$ can be computed prior to the operation and stored in the memory. Therefore, we neglect the computational complexity due to the mentioned precalculations. Consequently, the total number of operations performed by IGP to compute $H^{\mathbf{f}}_{k,l}$ for all measurements at time k scale with $\mathcal{O}(nm^2)$ using a naive matrix multiplication. Additionally, the formulation of IGP requires to compute the measurement covariance matrix $R_{k,l}$ for each measurement individually as indicated in (A.13) and (A.14). This procedure is dominated by the computation of $R^{\mathbf{f}}_{k,l}$

$$R^{\mathbf{f}}_{k,l} = k(\theta^{L}_{k,l}, \theta^{L}_{k,l}) - H^{\mathbf{f}}_{k,l}\ k(\boldsymbol{\theta}^{\mathbf{f}}, \theta^{L}_{k,l}) \ ,$$

Table 5.1: Disparity in the Computational Complexities of the Approaches at a Single Filter Recursion.

	Measurement Model Execution	**Measurement Noise Covariance Computation**
FGP	$\mathcal{O}(nm)$	–
IGP	$\mathcal{O}(nm^2)$	$\mathcal{O}(nm)$

where $k(\boldsymbol{\theta}^{\mathbf{f}}, \theta^L_{k,l}) \in \mathbb{R}^m$. The total computational complexity of this routine regarding all measurements is $\mathcal{O}(nm)$.

On the other hand, the measurement model of FPG entails to compute $\boldsymbol{\varphi}_{k,l}$ matrix in (5.26),

$$\boldsymbol{\varphi}_{k,l} \triangleq \boldsymbol{\varphi}(\theta^L_{k,l}) = [\phi_1(\theta^L_{k,l}) \ldots \phi_m(\theta^L_{k,l})]^\top,$$

where the basis functions $\{\phi_j(\cdot)\}_{i=1}^m$ are analytically available, hence we can construct $\boldsymbol{\varphi}_{k,l}$ by simply evaluating the basis functions at the corresponding input point. The computational complexity of this procedure is $\mathcal{O}(nm)$ for all measurements. Moreover, in FGP, the measurement covariance matrix R is identical for all measurements, thus it does not require any additional operations. The considered computational complexities of the methods is summarized in Table 5.1.

As a numeric example, we consider the 2D tracking setting where the methods make use of 50 variables to describe the target extent, and the number of measurement per frame is 10. The number of computations required for the abovementioned operations per frame is obtained to be 25×10^3 and 500 for IGP and FGP, respectively. For the 3D case, where the methods use 1000 extent variables, and the number of measurements per frame is 10, the number of computations for IGP and FGP is 10^7 and 10^4, respectively.

Notice that the above analysis is built upon the assumption that both of the methods utilize the same number of variables to describe the latent extent. However, in the subsequent sections, it will be shown that FGP can achieve the same tracking performance with IGP by using considerably fewer extent variables. Therefore, the computational advantage of FGP will be further highlighted in this regard. Furthermore, we hereby assume that during the measurement update phase of the employed inference schemes, each measurement will be processed for only one time. However, alternative techniques, such as the iterated EKF, or the variational Bayesian methods, requires the filter to process the measurements for multiple iterations. In such as a scheme, the computational complexity of the regarded operations will scale proportionally, hence the proposed approach will offer further advantages.

5.6.2 Experiments with Simulated Data

5.6.2.1 2D Tracking Problem

In this section, we investigate the performance of the proposed approach for the 2D EOT problem using simulated data. The conducted experiments consider a target traversing a path, which involves straight lines merged by a u-turn, at a constant speed v as illustrated in Fig. 5.3. We examine three different settings that present varying degrees of difficulty for a tracking application. Experiment 1 represents a typical tracking scenario, where the uncertainty in the target's motion is tolerable, and the sensor acquires measurements at a sufficiently high frame rate. In Experiment 2, the target moves along the same path at a higher speed thus exhibits a more agile maneuver during the u-turn. In Experiment 3, the frame rate of the sensor is decreased, hence the measurements are acquired less frequently. For all experiments, we examine three different extent models: a rectangle (T1), a triangle (T2), and a random extent (T3) that is sampled from the GP prior. The values of the sampling time T and the velocity v used in each experiment are reported in Table 5.2. In the simulations, the point measurements are generated by randomly sampling the target contour and perturbing by an independent sensor noise, which is Gaussian with zero mean and covariance $R = 0.3^2 I_2$ m^2. At each sensor scan, the number of measurements is randomly drawn from a Poisson distribution with a mean 15.

We examine the performance of the EOT algorithm introduced in Section 5.3.1, which is denoted as 'FGP-2D', in comparison to the reference method 'IGP-2D' [26]. Both of the mentioned trackers employ an EKF for inference. We also consider another

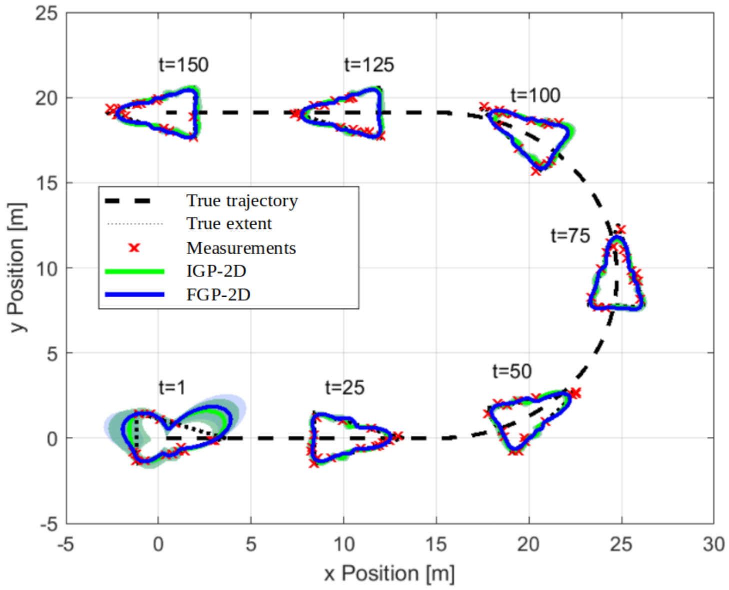


Figure 5.3: A typical realization of the u-turn experiment for a triangular object.

Table 5.2: Parameters Used for the 2D EOT Experiments with Simulated Data

	Simulation Parameters		**Filter Parameters**	
	v [m/s]	T [s]	σ_c [m]	σ_ψ [rad]
Exp1	0.5	1	0.1	0.01
Exp2	2	1	0.5	0.05
Exp3	0.5	4	0.1	0.01

reference method dubbed as ‘VB’ [35]. VB regards the same state-space model with [26] relying on the inducing point approximation of the GP model. However, it computes the posterior of the state vector by an alternative mechanism based on the variational Bayesian technique. In particular, VB performs multiple variational iterations through the available measurements in a single measurement update of the filter. To be able assess comparatively with VB, we realize a variant of the suggested method, indicated as ‘FGP-2D*’ that adopts an iterated EKF for inference. Similarly to VB, FGP-2D* benefits from multiple iteration for the measurement update.

For a fair comparison, all trackers are provided with the same process model, which includes the nearly constant velocity model for the kinematics in (5.29), and the dynamics of the extent is described by the model given in (5.30). The utilized process noise standard deviations for the center σ_c and orientation σ_ψ in three experiments are revealed in Table 5.2. Additionally, all methods employ the same GP model with zero mean function and the periodic covariance function [26], defined by

$$k\left(\theta,\theta'\right) = \sigma_f^2 e^{-\frac{2\sin^2\left(\frac{(\theta-\theta')}{2}\right)}{l^2}} + \sigma_r^2\ , \tag{5.49}$$

where σ_f^2 is the prior variance, l is the length-scale, and σ_r^2 is the constant term. The hyperparameters of the GP model are set to $\sigma_f = 2$ m, $\sigma_r = 0.6$ m and $l = \pi/6$ rad; and the number of inducing points is 50, which are evenly spaced in $[0, 2\pi]$. $\lambda = 0.99$ is used for the extent dynamics. The number of the iterations for VB and FGP-2D* is set to 10 for all experiments.

To evaluate the performance of the trackers, we regard the Intersection-Over-Union (IOU) measure, which jointly account for both the accuracy of kinematic and the extent estimates. The IOU is defined by

$$\text{IOU}(\hat{S}, S) = \frac{\text{area}(\hat{S} \cap S)}{\text{area}(\hat{S} \cup S)}, \tag{5.50}$$

where the extent of estimated object is represented by $\hat{S}$, and the true extent is represented by S. IOU$(\hat{S}, S) \in [0, 1]$ where 1 corresponds to perfect overlap between the estimated and the true extent while 0 implies these are disjoint. Furthermore, we consider the root-mean-square error (RMSE) of the orientation angle as a complementary performance measure,

$$\text{RMSE}(\hat{\psi}) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(\hat{\psi}_k - \psi_k)^2}\ , \tag{5.51}$$

where $\hat{\psi}$ and ψ indicate the estimated and the true orientation angle, respectively.

The mean of the IOU values and the RMSE of the target orientation for all cases are given in Table 5.3. All numbers are averaged over 100 Monte Carlo (MC) runs with randomly generated measurements (for all experiments) and target extents (T3). The results suggest that all methods are equally successful in Experiment 1. However, under the challenging conditions of Experiment 2 and 3, we can categorize the trackers into two: Benefiting from multiple iterations in a single measurement update, VB and FGP-2D* outperform the other methods, and they perform comparably in the

Table 5.3: Average IOU and Orientation RMSE Values of the 2D EOT Algorithms in the Experiments with Simulated Data

		IOU			**RMSE($\hat{\psi}$) [rad]**		
		T1	**T2**	**T3**	**T1**	**T2**	**T3**
Exp1	**IGP-2D**	0.878	0.871	0.906	0.086	0.048	0.067
	FGP-2D	0.869	0.860	0.904	0.092	0.042	0.061
	FGP-2D*	0.881	0.868	0.909	0.082	0.039	0.068
	VB	0.882	0.863	0.906	0.075	0.045	0.059
Exp2	**IGP-2D**	0.588	0.566	0.689	1.107	0.739	0.736
	FGP-2D	0.666	0.617	0.744	0.798	0.612	0.403
	FGP-2D*	0.824	0.752	0.879	0.150	0.330	0.084
	VB	0.834	0.780	0.827	0.129	0.225	0.263
Exp3	**IGP-2D**	0.585	0.561	0.675	1.232	0.767	0.669
	FGP-2D	0.661	0.620	0.740	0.845	0.750	0.560
	FGP-2D*	0.814	0.761	0.877	0.194	0.405	0.095
	VB	0.829	0.770	0.853	0.179	0.281	0.114

IOU and the orientation RMSE. On the other hand, in these experiments, FGP-2D is observed to be more robust than IGP for all extent models.

Additionally, we attempt to directly observe the added value of the proposed formulation based on the reduced-rank GP approximation in the context of extent estimation. For this purpose, we conduct an experiment where the target is fixed throughout the scenario. Accordingly, we exclude the kinematic variables from the state vector, and exclusively focus on the extent estimation performances of FGP and IGP. In each simulation run, the target extent is randomly sampled from the prior GP with the periodic kernel function. The duration of the experiment 10 s, where the sampling time is set to 1 s. In each time instant, we acquire 10 newly available observation from the target. This experiment is repeated for different number of extent variables for FGP and IGP, and the averaged IOU values over 100 MC runs are reported in Fig. 5.4. It is shown that both algorithms perform successfully for large number of extent variables as they attain high IOU scores. On the other hand, FGP is observed to exhibit more robust performance for fewer extent variables as it leads to higher IOU values in this setting.

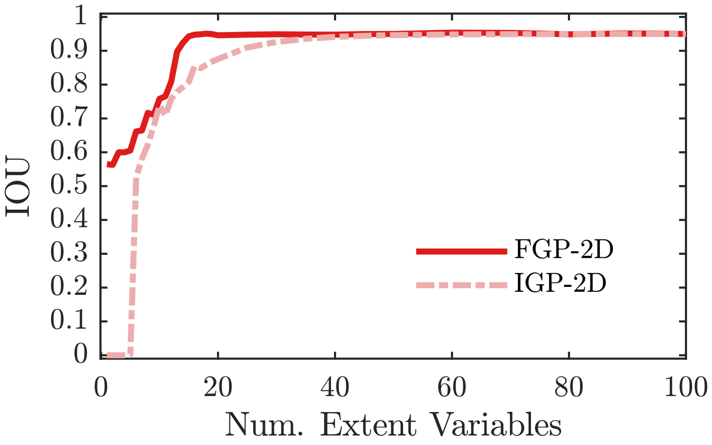
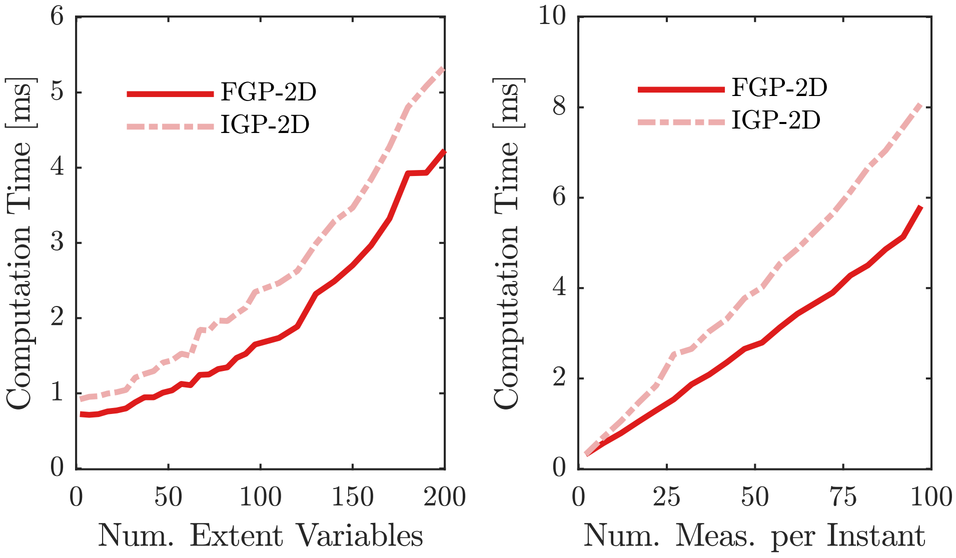


Figure 5.4: Average IOU plots of FGP and IGP obtained for different number extent variables.



(a)

(b)

Figure 5.5: Average computation time for an iteration of the 2D EOT algorithms. (In Fig. (a), the number of available measurements is set to 10. In. Fig. (b), the number of extent variables is set to 20.)

To empirically investigate the computational characteristics of FGP and IGP, we compute the average computation time of a single filter recursion for both methods with respect to the number of extent variables and the number of available measurements. The findings averaged over 100 MC runs are shown in Fig. 5.5. The average computational time of FGP scales in a more preferable fashion in regard to both the number of extent variables and the number of measurements per frame.

As a final remark, Fig. 5.4 suggests that FGP and IGP achieve 90% of their best IOU performance with 15 and 30 extent variables, respectively. Considering the case where 10 measurements are provided to the methods per sensor scan, a single recursion of the trackers takes nearly 0.7 ms and 1.3 ms for FGP and IGP, respectively.

5.6.2.2 3D Tracking Problem

In this section, the performance of the proposed approach is evaluated for the 3D EOT problem over simulated data. To present the results in a comparative manner, we regard the algorithm introduced in [34], which uses the inducing point approximation of the underlying GP model. Throughout this section, we will refer to our method as 'FGP-3D', while 'IGP-3D' will stand for the reference approach.

We investigate two distinct scenarios, which are indicated by S1 and S2. In S1, the target moves along a straight line at a constant speed of 0.5 m/s. In S2, the target follows a curved path while performing combined rotations around different axes. For all experiments, we examine four different extent models: a sphere (T1), an ellipsoid (T2), a box (T3), and a cone (T4). The dimensions of the objects are as follows: the radius of the sphere is 3 m, the semi-axes of the ellipsoid are $(2.5, 1, 1)$ m in length, the length of the edge of the cube is 3 m, and the base radius and height of the cone are 1.5 m and 4 m, respectively. The point measurements are generated from random sources on the target surface and perturbed by independent Gaussian noise with zero mean and covariance $0.1^2 I_3$. At each sensor scan, 20 point measurements are originated from random sources which are sampled from a uniform distribution defined over the object surface. The measurements are produced at 10 Hz, hence the sampling time of all algorithms is set to $T = 0.1$ s. We assess the performance of the trackers through the IOU measure.

Both of the methods utilize the same process model detailed in Section 5.4.2.2, and they regard the standard squared exponential covariance function for the GP model,

$$k(\mathbf{p}, \mathbf{p}') = \sigma_f^2 e^{-\frac{\|\mathbf{p}-\mathbf{p}'\|^2}{2l^2}} + \sigma_r^2. \tag{5.52}$$

The process noise standard deviations for the kinematics are set to $\sigma_c = 0.1$ m and $\sigma_\alpha = 0.1$, and $\lambda = 0.99$ is used for the extent dynamics; the hyperparameters of the GP model are set to $\sigma_f = 1, \sigma_r = 0.2, l = \pi/8$; the inducing points used by IGP-3D to represent the extent are evenly spaced in the spherical coordinates. All experiments are repeated 100 times with random realizations of the measurement noise and the measurement origins. The averaged IOU values are given in Table 5.4. For all of the settings, both of the methods successfully estimate the extent, and their performance is comparable in each case.

Moreover, we conduct an experiment to analyze the two alternative approaches specifically in the context of extent estimation. Throughout the scenario, a cube-shaped object is kept stationary at a known position. Accordingly, similar to the 2D setting, the kinematic variables are excluded from the state vector, hence both FGP-3D and IGP-3D exclusively estimate the unknown object extent. The duration of the experiment is 10 s, and the sampling time is set to 0.1 s. The number of point measurements per instant is 20. The experiment is repeated for different number of extent variables for FGP-3D and IGP-3D, and Fig.5.6 reveals the averaged IOU values over 1000 MC runs. Both methods are able to produce satisfactory extent estimates when there is a large number of extent variables. By contrast, FGP-3D is significantly superior to its counterpart for relatively fewer extent variables.

Table 5.4: Average IOU Values of the 3D EOT Algorithms in the Experiments with Simulated Data

	S1				**S2**			
	T1	**T2**	**T3**	**T4**	**T1**	**T2**	**T3**	**T4**
IGP-3D	0.971	0.908	0.906	0.846	0.972	0.893	0.902	0.854
FGP-3D	0.975	0.920	0.912	0.846	0.975	0.893	0.916	0.855

To examine the computational characteristics of the methods, we compute the average computation time of a single filter recursion with respect to the number of extent variables and the number of processed measurements per frame. We repeat the experiments for 1000 times and the averaged numbers are depicted in Fig. 5.7. FGP-3D scales in a more favorable manner with respect to both the number of extent variables and the number of measurements per frame.

5.7 Conclusion and Discussions

In recent years, there have been many attempts to construct flexible models to tackle the EOT problem. In this context, the GPEOT framework has attracted a lot of attention in the research community. This approach provides a convenient basis to recursively estimate the unknown target extent together with the kinematic state, which includes position, velocity and orientation, by processing 2D and 3D point cloud data. In this chapter, we suggest an alternative formulation for the GPEOT algorithms. More specifically, we adopt a spectral-domain approximation for the underlying GP model that describes the target extent. The utilized approximation forms a basis function expansion, the weights of which provide a parametrized representation of the latent extent. We derive state-space models regarding the extent and the kinematic variables for 2D and 3D tracking applications. Subsequently, we efficiently estimate the state vector employing standard Bayesian inference techniques, which in turn constitutes an EOT approach. It is shown that the resulting trackers significantly reduce the computational load with respect to their existing counterparts and lead to more robust performance with small number of extent variables.

The enhanced computational characteristics of the introduced approach will bring added value for many practical applications. In particular, autonomous mobile platforms are required to achieve a reliable perception of their surrounding and make decisions accordingly in real-time. In this pursuit, they need to carry out computationally intensive processes, for example, to extract information from raw data collected via multiple sensor devices, while the computational resources are severely limited. The presented methods will enable attaining precise knowledge of the dynamic elements in the environment at a lower computational load. In this respect, we believe that the proposed algorithms will contribute to the adoption of the GPEOT models for the mentioned applications.

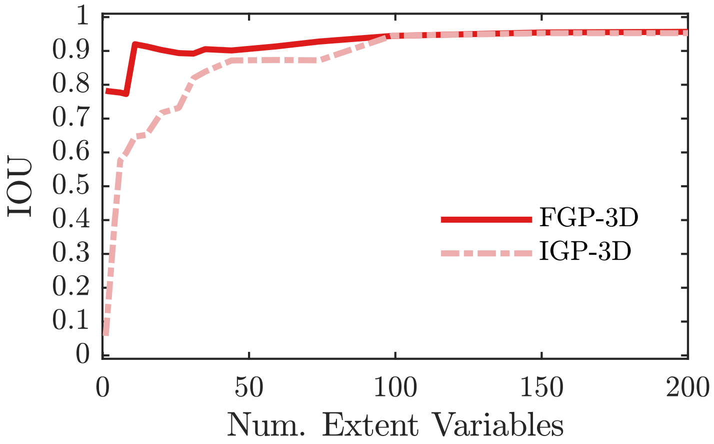
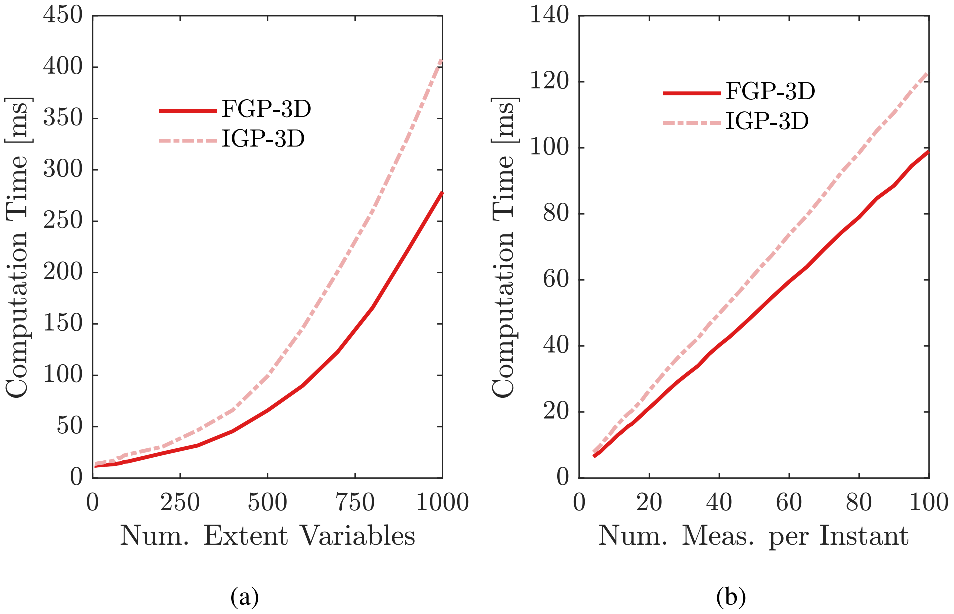


Figure 5.6: Average IOU plots of FGP-3D and IGP-3D obtained for different number extent variables.



(a)

(b)

Figure 5.7: Average computation time for an iteration of the 3D EOT algorithms. (In Fig. (a), the number of available measurements is set to 20. In. Fig. (b), the number of extent variables is set to 162.)

CHAPTER 6

TRACKING ARBITRARILY-SHAPED EXTENDED TARGETS USING GAUSSIAN PROCESSES

6.1 Introduction

Detailed shape estimates offer a great value for many applications, for instance, automated mobile platforms, which are typically expected to operate in dynamic and unstructured environments, need to have refined knowledge of the occupied and free regions in their surrounding for effective and efficient navigation. Additionally, shape estimates deliver valuable cues on the objects' characteristics, which might prove useful to identify the object type and to anticipate its future behavior.

In the context of object tracking, a high precision extent model enables to accurately express the relation between observations and the object shape, which in turn potentially increases the tracking performance. Moreover, it is also beneficial to realize successful measurement association. In the literature of extended object tracking, there have been many attempts that strive for more flexible extent representations that may apply to a wider class of objects. In this line of research, the arguably most popular approach relies on a radial function description of the underlying object extent [26], [13]. This formulation assumes that the object of interest is star-convex, and it clearly introduces some limitations from a practical point of view. With this in mind, in this chapter we propose an algorithm that simultaneously estimates the kinematics and learns an arbitrarily-shaped object's extent without imposing any convexity assumption.

Seeking for a general extent description, we hereby suggest to model the unknown object extent by a scalar-valued potential function. The potential value of a point in space is simply used to indicate whether the point is included by the object extent or not. With this formulation, it is possible to represent arbitrarily-shaped objects, which might be non-convex and consisting of disconnected subparts. The rest of the derivation closely follows the existing GPEOT models. In particular, we model the latent potential function by a GP, which is approximated at some inducing points. Subsequently, a unified state-space model, which consists of both the kinematic variables and the approximate description of the GP model, is constructed. Regarding this state-space model, we implement an EKF that efficiently estimates the state vector by recursively processing point cloud measurements. The performance of the proposed algorithm is demonstrated through simulation experiments.

6.2 Proposed Method

6.2.1 Extent Representation

In this section, we will reveal the representation of the object extent used in the derivation. While formulating the description, our objective is to attain high representational power, which will allow the resulting method to generalize to a broad class of object shapes, and be sufficiently compact for an online tracking application.

In this regard, we model the object extent by a scalar potential function, i.e., $f(\mathbf{u}) : \mathbb{R}^d \to \mathbb{R}$, where $\mathbf{u} \in \mathbb{R}^d$ indicates a point in the coordinate frame. More specifically, we follow the conventional denotation and define the potential function as

$$f(\mathbf{u}) = \begin{cases} +1, & \mathbf{u} \in S \\ -1, & \mathbf{u} \notin S \end{cases}, \tag{6.1}$$

where S denotes the set of all points contained within the object extent. $f(\mathbf{u})$ can be simply interpreted as a descriptor function whose level-sets imply the latent object extent. The same potential function has been widely utilized to characterize the occupancy information in the applications of environmental exploration and mapping, see, for example, [97], [98], [99], [100], [101].

This extent model is of critical importance for the derivation of a tracking algorithm that can learn the extent of the arbitrarily-shaped objects. The model does not impose any assumption upon the shape of the object, such as star-convexity. It can even express an object extent consisting of multiple disconnected subparts.

In the upcoming sections, we will first introduce a probabilistic model for the unknown potential function, which later will be translated into a unified state-space model for joint estimation of the kinematics and the extent of the object.

6.2.2 Gaussian Process Modeling of Object Extent

Being a non-parametric model, GP establishes a convenient basis for probabilistic modeling of arbitrary potential functions and hence the corresponding object extents. In this regard, $f(\mathbf{u})$ is assumed to have a GP prior, i.e., $f(\mathbf{u}) \sim \mathcal{GP}(0, k(\mathbf{u}, \mathbf{u}'))$, where $k(\mathbf{u}, \mathbf{u}')$ is the covariance function, which basically computes the covariance between the function evaluations at locations $\mathbf{u}$ and $\mathbf{u}'$.

In a typical object tracking setting, the problem is to estimate the latent variables by processing s which become available in a sequential fashion. As the number of measurements monotonically increases over time, a naive implementation of a GP model is infeasible due the ever-growing storage and computational requirements. Consequently, we hereby employ a recursive approximation of the GP model, which brings in favorable computational properties.

6.2.3 Recursive Gaussian Process Regression

We rely on an approximation of a GP that summarizes the original model at a finite set of basis points, i.e., $\mathbf{f} \triangleq [f(u_1^{\mathbf{f}}) \, \dots \, f(u_{N^{\mathbf{f}}}^{\mathbf{f}})]^\top$ evaluated at the inputs $\mathbf{u}^{\mathbf{f}} \triangleq [u_1^{\mathbf{f}} \, \dots \, u_{N^{\mathbf{f}}}^{\mathbf{f}}]^\top$. More specifically, assuming $\mathbf{f}$ provides the sufficient statistics for the observations, we will be able to compute the posterior distribution $p(\mathbf{f}|y_{1:N})$ recursively. To this end, we refer to the underlying GP model to offer the measurement likelihood function and the prior density. The joint distribution of an arbitrary function value $f_k = f(u_k)$ and the basis points $\mathbf{f}$ is revealed by the GP as

$$\begin{bmatrix} f_k \\ \mathbf{f} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(u_k, u_k) & K(u_k, \mathbf{u}^{\mathbf{f}}) \\ K(\mathbf{u}^{\mathbf{f}}, u_k) & K(\mathbf{u}^{\mathbf{f}}, \mathbf{u}^{\mathbf{f}}) \end{bmatrix}\right). \tag{6.2}$$

Then, by the standard GP regression equations we can immediately offer the following,

$$p(f_k|\mathbf{f}) = \mathcal{N}(f_k; H_k^{\mathbf{f}}\mathbf{f}, R_k^{\mathbf{f}}), \tag{6.3a}$$

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{0}, P_0^{\mathbf{f}}), \tag{6.3b}$$

where

$$H_k^{\mathbf{f}} = H^{\mathbf{f}}(u_k) = K(u_k, \mathbf{u}^{\mathbf{f}})[K(\mathbf{u}^{\mathbf{f}}, \mathbf{u}^{\mathbf{f}})]^{-1}, \tag{6.3c}$$

$$R_k^{\mathbf{f}} = R^{\mathbf{f}}(u_k) = k(u_k, u_k) - K(u_k, \mathbf{u}^{\mathbf{f}})[K(\mathbf{u}^{\mathbf{f}}, \mathbf{u}^{\mathbf{f}})]^{-1}K(\mathbf{u}^{\mathbf{f}}, u_k), \tag{6.3d}$$

$$P_0^{\mathbf{f}} = K(\mathbf{u}^{\mathbf{f}}, \mathbf{u}^{\mathbf{f}}). \tag{6.3e}$$

Regarding (6.3), we can write the equivalent state-space model, which then can be used within a Kalman filter for recursive inference, [26].

$$\mathbf{f}_{k+1} = \mathbf{f}_k, \tag{6.4a}$$

$$f_k = H_k^{\mathbf{f}}\, \mathbf{f}_k + e_k^{\mathbf{f}}, \quad e_k^{\mathbf{f}} \sim \mathcal{N}(0, R_k^{\mathbf{f}}), \tag{6.4b}$$

$$\mathbf{f}_0 \sim \mathcal{N}(\mathbf{0}, P_0^{\mathbf{f}}), \tag{6.4c}$$

where $\mathbf{f}_k$ is defined as the latent function values evaluated at the predetermined inputs, i.e., $\mathbf{f}_k \triangleq \mathbf{f}$.

6.2.4 State-Space Model

We construct a state-space model considering both the kinematics and the extent of the object. The state vector is defined as $\mathbf{x}_k \triangleq \left[\bar{\mathbf{x}}_k^\top\ \mathbf{f}_k^\top\right]^\top$, where $\bar{\mathbf{x}}_k$ denotes the kinematic variables consisting of the object position $\mathbf{c}_k$, orientation ψ_k, and the corresponding time derivatives, $\mathbf{x}_k^*$, i.e., $\bar{\mathbf{x}}_k \triangleq \left[\mathbf{c}_k^\top\ \psi_k\ \mathbf{x}_k^{*\top}\right]^\top$; $\mathbf{f}_k$ indicates the parameterized description of the potential function.

6.2.4.1 Measurement Model

In this section, we derive a measurement model that reveals the connection between the state variables and the observations.

$\mathcal{Y}_k = \{y_{k,l}\}_{l=1}^{n_k}$ denotes the set of point measurements collected at time k. In this study, we assume that a point measurement $y_{k,l}$ encapsulates two types of information: (i) the detection point in space $\mathbf{p}_{k,l} \in \mathbb{R}^2$, (ii) the associated occupancy/free-space information $o_{k,l}$. We designate $o_{k,l}$ as a binary-valued variable, i.e., $o_{k,l} \in \{-1, +1\}$, in accordance with the definition of the potential function in (6.1). In particular, $o_{k,l}$ takes the value of $+1$ for occupancy measurements, or it is -1 for free-space measurements. Consequently, a single measurement is specified by the tuple $y_{k,l} = (\mathbf{p}_{k,l}, o_{k,l})$.

In practice, the considered measurements can be generated by various sensors modalities, such as laser range scanners, depth cameras, or by processing the images captured by monocular cameras. In all these cases, the measurements provide some noisy observations of the true source points that give rise to the measurements. In this respect, we relate $o_{k,l}$ to the underlying potential function evaluated at $\mathbf{p}^L_{k,l}$ as in

$$o_{k,l} = f(\mathbf{p}^L_{k,l}(\mathbf{c}_k, \psi_k)) + \bar{e}_{k,l}, \quad \bar{e}_{k,l} \sim \mathcal{N}(0, R) \tag{6.5}$$

$\bar{e}_{k,l}$ is assumed to be zero-mean Gaussian measurement noise with covariance R, and the expression for $\mathbf{p}^L_{k,l}$ is given as

$$\mathbf{p}^L_{k,l}(\mathbf{c}_k, \psi_k) = R(\psi_k)(\mathbf{p}_{k,l} - \mathbf{c}_k). \tag{6.6}$$

where $\mathbf{p}_{k,l}$ is the corresponding point resolved in the global coordinate frame, and $R(\psi_k)$ is the rotation matrix between the global and the local frames defined as

$$R(\psi_k) = \begin{bmatrix} \cos(\psi_k) & \sin(\psi_k) \\ -\sin(\psi_k) & \cos(\psi_k) \end{bmatrix}. \tag{6.7}$$

Then, by plugging in the GP representation of the potential function, we can rewrite the measurement equation as

$$\begin{aligned} 0 &= \underbrace{-o_{k,l} + H^{\mathbf{f}}\left(\mathbf{p}^L_{k,l}(\mathbf{c}_k, \psi_k)\right)\, \mathbf{f}_k}_{=h_{k,l}(\mathbf{x}_k, y_{k,l})} + \underbrace{e^{\mathbf{f}}_{k,l} + \bar{e}_{k,l}}_{=e_{k,l}} \\ &= h_{k,l}(\mathbf{x}_k, y_{k,l}) + e_{k,l}, \quad e_{k,l} \sim \mathcal{N}(0, R_{k,l}). \end{aligned} \tag{6.8a}$$

where

$$R_{k,l} = R^{\mathbf{f}}_{k,l} + R. \tag{6.8b}$$

We illustrate the variables used in the measurement model in Fig. 6.1. Notice that this is a nonlinear implicit model, where the measurement can not be explicitly written as function of the state variables and the noise term. This type of models have been commonly used in the literature to derive recursive Bayesian filters for various applications, such as localization and tracking [12, 65, 31, 29].

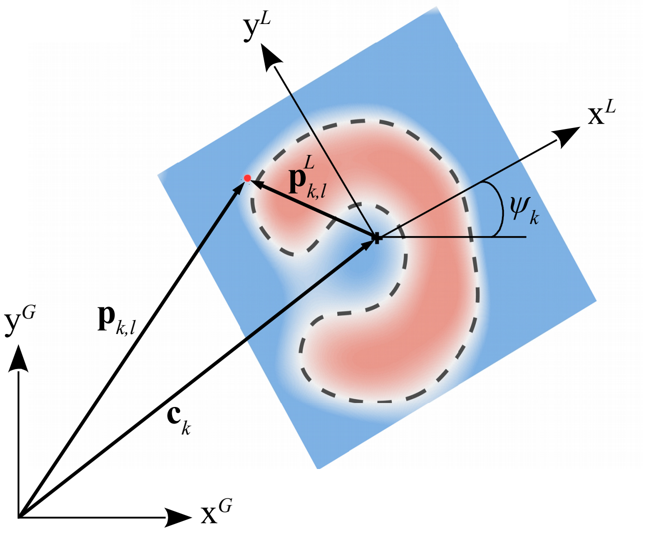


Figure 6.1: An illustration of the variables included in the measurement model. The red circle indicates the point measurement. The dashed curve denotes the object of interest.

6.2.4.2 Process Model

To complete the definition of the state-space model, we formulate the time evolution of the state vector by the following process model.

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q_k), \tag{6.9}$$

where F is the system matrix, and $\mathbf{w}_k$ is the process noise, which is assumed to be Gaussian with zero mean and covariance Q_k. Suppose that the dynamics of the kinematic and the extent variables do not interact with each other, then we can partition the process model as

$$F = \begin{bmatrix} \bar{F} & 0 \\ 0 & F^{\mathbf{f}} \end{bmatrix}, \, Q_k = \begin{bmatrix} \bar{Q} & 0 \\ 0 & Q_k^{\mathbf{f}} \end{bmatrix}. \tag{6.10}$$

The formulation does not put any restriction on the selection of the process model. We hereby employ the nearly constant velocity model for the object kinematics, i.e.,

$$\bar{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes I_3, \; \bar{Q} = \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \otimes \begin{bmatrix} \sigma_c^2 & 0 & 0 \\ 0 & \sigma_c^2 & 0 \\ 0 & 0 & \sigma_\psi^2 \end{bmatrix}, \tag{6.11}$$

where σ_c^2 and σ_ψ^2 are the process noise variances for the position and the orientation, respectively; T is the sampling time; and $\otimes$ denotes the Kronecker product.

For the object extent, we utilize the following dynamic model, which is known to attain the predicted distribution with maximum entropy for unknown and slowly varying processes [59, Therorem 1],

$$F^{\mathbf{f}} = I, \; Q_k^{\mathbf{f}} = \left(\lambda^{-1} - 1\right) P_{k|k}^{\mathbf{f}}, \tag{6.12}$$

where I indicates the identity matrix, $P_{k|k}^{\mathbf{f}}$ is the covariance of the estimated extent. Notice that this model basically scales up the prediction covariance as $P_{k+1|k}^{\mathbf{f}} = \lambda^{-1} P_{k|k}^{\mathbf{f}}$ for $\lambda < 1$, while the mean of the prediction is the same with that of the estimated density.

6.2.4.3 Inference

In this section, we will present an effective inference method by relying on the state-space model developed earlier. Particularly, the objective is to propose a method that recursively computes the posterior distribution of the state vector. To this end, we will utilize an extended Kalman filter (EKF) considering the nonlinearities in the measurement model.

To incorporate multiple measurements $\{y_{k,l}\}_{l=1}^{n_k}$ available at time k, we first form a measurement vector by concatenating the individual measurements,

$$\mathbf{y}_k = \left[y_{k,1}, \; \ldots, \; y_{k,n_k}\right]^\top. \tag{6.13}$$

Then, the corresponding measurement model can directly be written as

$$\mathbf{0} = \mathbf{h}(\mathbf{x}_k, \mathbf{y}_k) + \mathbf{e}_k, \quad \mathbf{e}_k \sim \mathcal{N}(\mathbf{0}, R_k), \tag{6.14}$$

where

$$\mathbf{h}(\mathbf{x}_k, \mathbf{y}_k) = [h(\mathbf{x}_k, y_{k,1}),\ \ldots, h(\mathbf{x}_k, y_{k,n_k})]^\top, \tag{6.15a}$$

$$\mathbf{y}_k = \left[y_{k,1}^\top,\ \ldots,\ y_{k,n_k}^\top\right]^\top, \tag{6.15b}$$

$$\mathbf{e}_k = \left[e_{k,1}^\top,\ \ldots,\ e_{k,n_k}^\top\right]^\top, \tag{6.15c}$$

$$R_k = \text{blkdiag}(R_{k,1}, \ldots, R_{k,n_k}), \tag{6.15d}$$

The given covariance matrix R is in block diagonal form, which follows from the assumption that the measurement noise terms are mutually independent.

In consequence, we obtain the following the state-space model by defining a Gaussian prior density for the state vector,

$$\begin{aligned} \mathbf{x}_{k+1} &= F\mathbf{x}_k + \mathbf{w}_k, & \mathbf{w}_k &\sim \mathcal{N}(\mathbf{0}, Q_k), \\ \mathbf{0} &= \mathbf{h}(\mathbf{x}_k, \mathbf{y}_k) + \mathbf{e}_k, & \mathbf{e}_k &\sim \mathcal{N}(\mathbf{0}, R_k), \\ \mathbf{x}_0 &\sim \mathcal{N}(\boldsymbol{\mu}_0, P_0). \end{aligned} \tag{6.16}$$

The EKF regards this model to compute the posterior distribution by processing the measurements collected at time k. The complete details of the filtering equations are given below.

- *Measurement update equations:*

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{0} - \hat{\mathbf{y}}_{k|k-1}) \tag{6.17a}$$

$$P_{k|k} = P_{k|k-1} + K_k H_k P_{k|k-1} \tag{6.17b}$$

  where

$$\hat{\mathbf{y}}_{k|k-1} = \mathbf{h}_{k,l}(\hat{\mathbf{x}}_{k|k-1}) \tag{6.17c}$$

$$K_k = P_{k|k-1} H_k^\top S_k^{-1} \tag{6.17d}$$

$$S_k = H_k P_{k|k-1} H_k^\top + R_k \tag{6.17e}$$

$$H_k = \frac{d}{d\mathbf{x}_k} \mathbf{h}_{k,l}(\mathbf{x}_k)|_{\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1}} \tag{6.17f}$$

- *Time update equations:*

$$\hat{\mathbf{x}}_{k+1|k} = F\hat{\mathbf{x}}_{k|k}, \tag{6.18a}$$

$$P_{k+1|k} = FP_{k|k}F^\top + Q. \tag{6.18b}$$

6.2.5 Disclosing the Object Extent

Our model relies on the potential function representation that inherently characterizes the underlying object extent, where the proposed inference scheme provides a complete probabilistic description with the estimated mean and the associated covariance. That being said, converting this description to an explicit definition of the object extent might be of particular interest for various purposes. For example, a detailed extent estimate might prove invaluable in path planning for an autonomous vehicle aiming to avoid obstacles, or for efficient grasp planning of a robot arm.

Let us define a binary class indicator c_* for the query point $\mathbf{u}_* \in \mathbb{R}^2$ such that $c_* = 1$ denotes that $\mathbf{u}_*$ is contained within the object extent, i.e., $\mathbf{u}_* \in \mathcal{S}$; while $c_* = 0$ indicates that $\mathbf{u}_*$ is a free-space point. To compute the corresponding class probabilities conditioned on the measurements collected up to time k, we will basically regard the probability density of the potential function at $\mathbf{u}_*$ conditioned on the acquired measurements, i.e., $p(f_*|\mathcal{Y}_{1:k})$ where $f_* \triangleq f(\mathbf{u}_*)$. To this end, we can immediately write

$$f_* = H_*^{\mathbf{f}} \mathbf{f}_k + e^{\mathbf{f}}, \;\; e^{\mathbf{f}} \sim \mathcal{N}(\mathbf{0}, R^{\mathbf{f}}) \tag{6.19}$$

where $H_*^{\mathbf{f}} \triangleq H^{\mathbf{f}}(\mathbf{u}_*)$, and using the posterior of the potential function representation computed by the EKF, $p(\mathbf{f}_k|\mathcal{Y}_{1:k}) = \mathcal{N}(\mu^{\mathbf{f}}_{k|k}, P^{\mathbf{f}}_{k|k})$, we obtain

$$\begin{aligned} &p(f_*|\mathcal{Y}_{1:k}) = \mathcal{N}(\mu_*, \sigma_*^2), \\ &\text{where } \mu_* = H_*^{\mathbf{f}} \mu^{\mathbf{f}}_{k|k}, \;\; \sigma_*^2 = H_*^{\mathbf{f}} P^{\mathbf{f}}_{k|k} {H_*^{\mathbf{f}}}^\top + R^{\mathbf{f}} \; . \end{aligned} \tag{6.20}$$

Subsequently, we follow the standard approach to obtain the class probabilities by squashing the estimate through a response function, [42, Ch. 3]. For the response function, we utilize the following cumulative density function (cdf) of the standart normal distribution, which in turn leads to a *so-called* probabilistic least-squares classifier [42, Ch. 6.5.1].

$$p\left(c_* = 1 \mid \mathcal{Y}_{1:k}\right) = \Phi\left(\frac{\alpha\mu_* + \beta}{\sqrt{1 + \alpha^2\sigma_*^2}}\right) \tag{6.21}$$

Φ is the cdf of the standard normal distribution; and it is used as a sigmoid function with parameters α and β. Accordingly, we can calculate the other class' probability as $p\left(c_* = 0 \mid \mathcal{Y}_{1:k}\right) = 1 - p\left(c_* = 1 \mid \mathcal{Y}_{1:k}\right)$. Finally, after the class probabilities are computed, it is possible to recognize the query point as one of the classes $\{$extent, free-space, unknown$\}$ by simply specifying some threshold values.

6.3 Simulation Results

In this section, we demonstrate the performance of the proposed method via simulation experiments. We investigate three different scenarios with different motion patters and sensor characteristics.

In the first experiment, the object of interest is static throughout the scenario. At each time instant, we uniformly sample the test environment and collect 20 measurements from the surface of the object and 50 measurements from the surrounding free-space region. The detection points of the measurements are corrupted by i.i.d. Gaussian noise with covariance $0.03^2 I_2 \; m^2$. The sampling time is used as $T = 0.1$ second, the total duration of the experiment is 10 seconds. We examine three different type of extents, i.e., a C-shape, a T-shape, and another one containing disconnected subparts. The tracker maintains the extent representation over 1024 basis points, which are equidistantly located on a grid over $[-3, 3] \; m \times [-3, 3] \; m$. The representative results obtained at four time instants are shown in Fig. 6.2.

In the second experiment, we investigate an object moving along a linear path at a constant speed of $1 \; m/s$. In this case, the observations are simulated to be acquired by a two-dimensional laser range scanner. The sampling time is used as $T = 0.1 \; s$, the total duration of the experiment is $10 \; s$. At each instant, the sensor emits 200 laser beams and reports the detection points of the ones reflected from the contour of the object. Additionally, for each each laser beam we produce 3 free-space measurements by randomly sampling the free section of the beam. All measurements are perturbed by i.i.d. Gaussian noise with covariance $0.01^2 I_2 \; m^2$. The results are exhibited for six distinct time instants in Fig. 6.2.

In the third experiment, we consider an object performing rotational motion. The measurements are simulated to be generated by a two-dimensional laser range scanner, whose characteristics are identical with the one utilized in the second experiment.

The results are reported in Fig. 6.4.

In all cases, the proposed method satisfactorily estimates the kinematic variables of the objects and learns the unknown potential function. In the first experiment, the information provided by the uniform measurements is successfully incorporated, hence at the end of the scenario, the uncertainty over the complete region is appropriately reduced, and the latent extent can easily be distinguished from the estimated potential function values. In the second experiment, due to the characteristics of the sensor, some sections of the object are always self-occluded in accordance with the object-sensor geometry. As the object moves along the trajectory, some of the previously unseen parts are explored by the laser beams, and the algorithm properly decreases the uncertainty on the observed portion while a larger uncertainty is associated with the unobserved regions. Notice that as the measurements are exclusively generated by the object contour in this case, the interior points of the extent remain mostly uncertain at the end of the experiment. Furthermore, the third experiment shows that the developed algorithm can successfully estimate the varying orientation of the object. As the object rotates, the method learns the underlying non-convex shape and appropriately decreases the uncertainty of the observed regions.

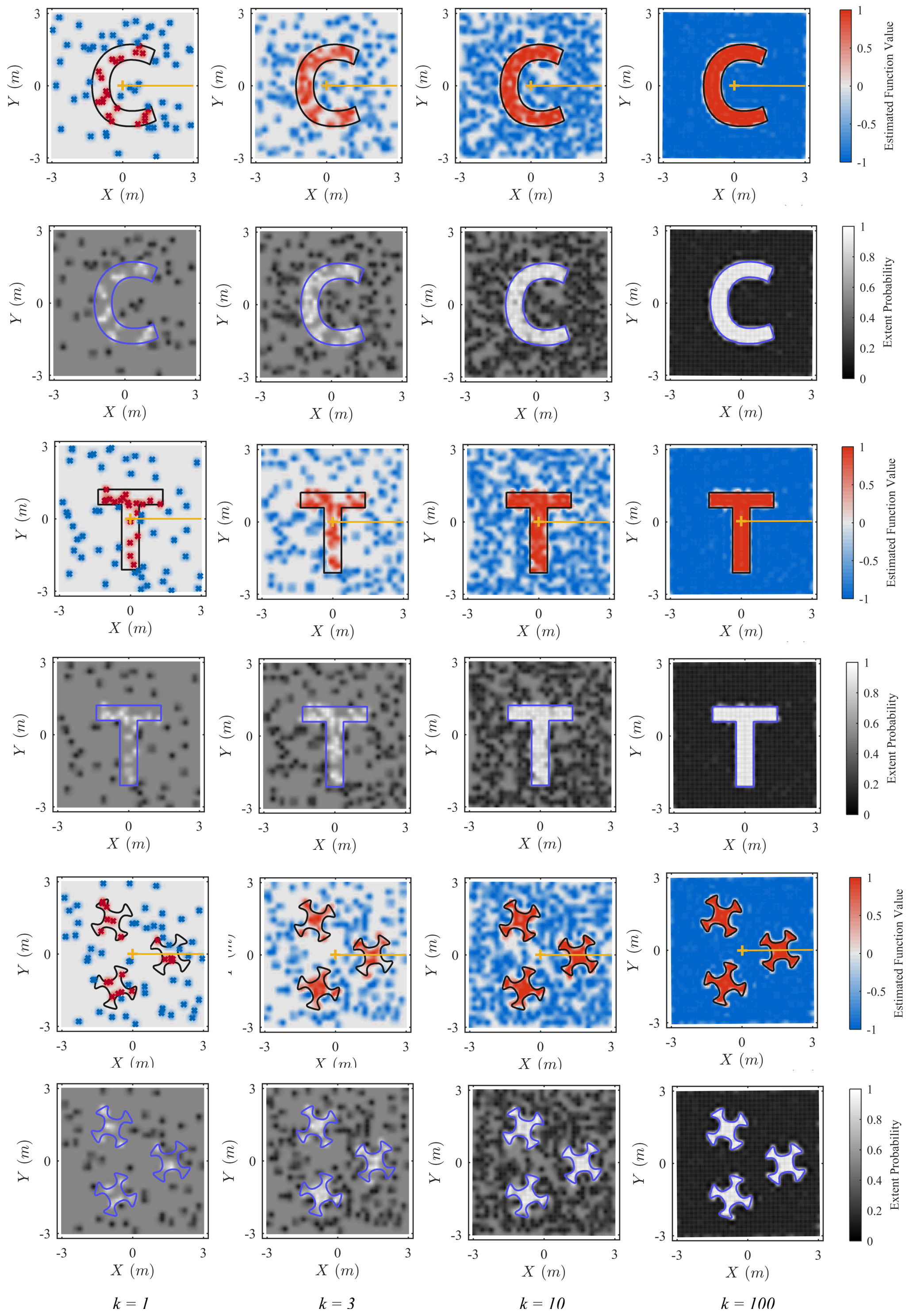


Figure 6.2: Results obtained at some representative time instants $k = \{1, 3, 10, 100\}$ during the first simulation experiment. The true extent of the object is visualized by the solid curve. The estimated center and the orientation are indicated by the yellow plus sign and the straight line, respectively. For the first instant, measurements originated from the extent and the free-space are shown by red and blue cross signs, respectively.

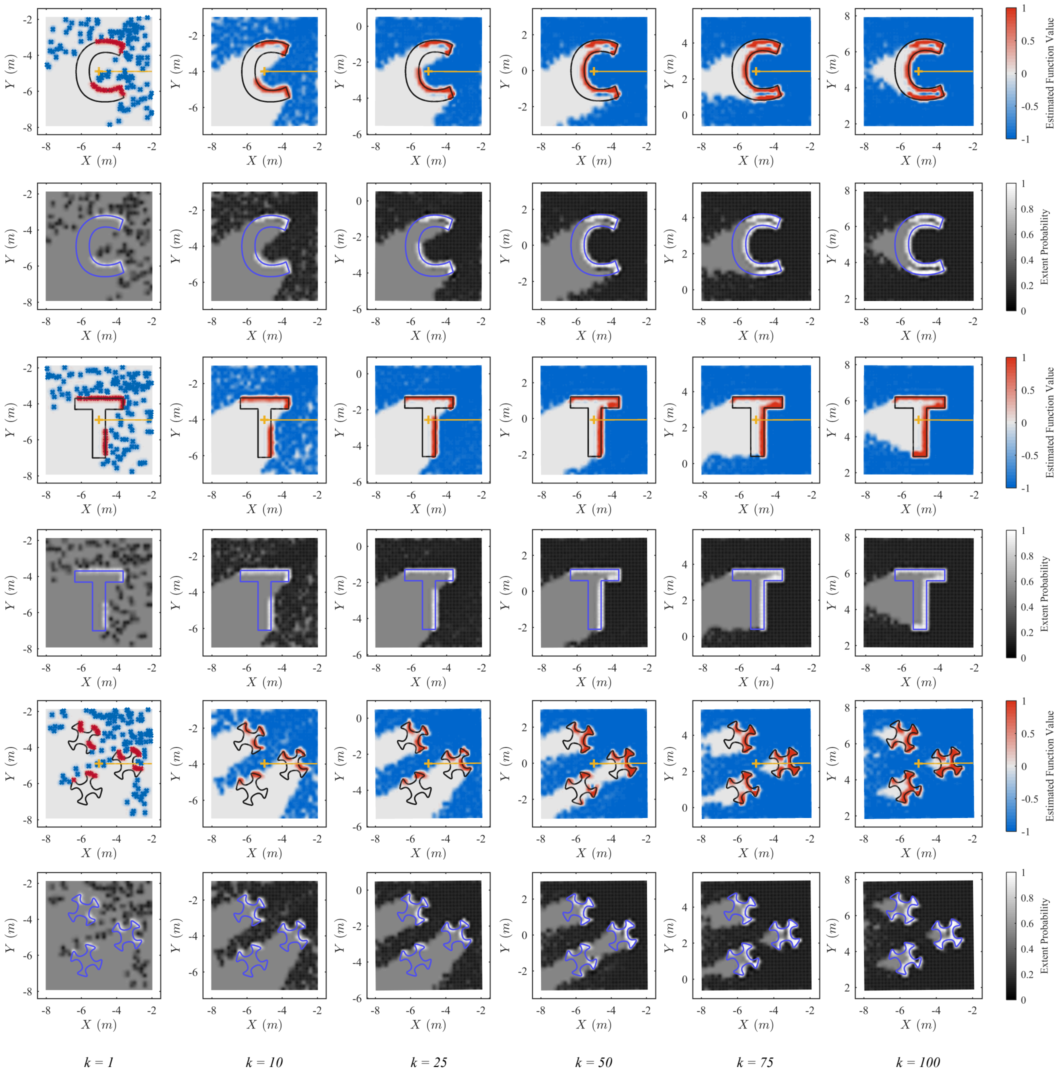
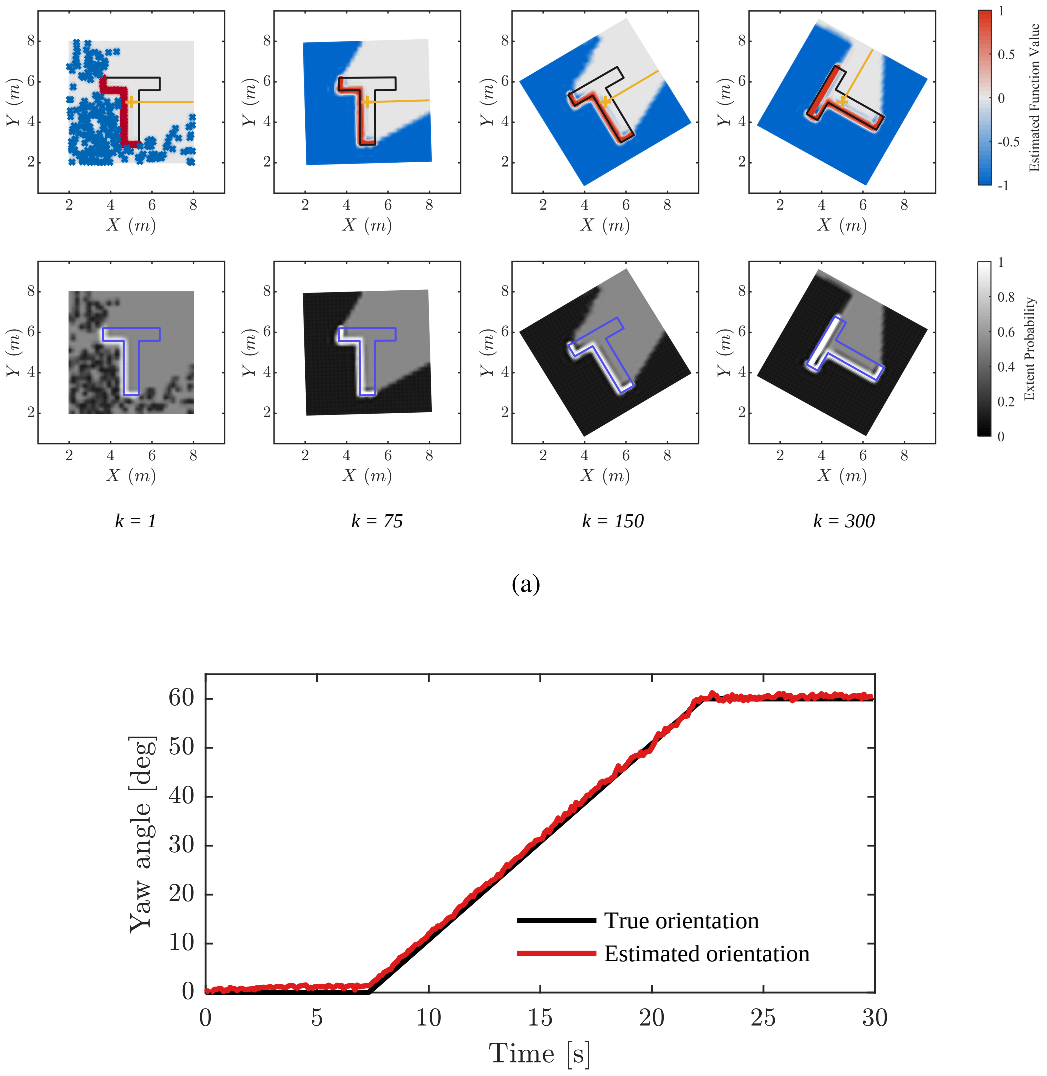


Figure 6.3: Results obtained at some representative time instants $k = \{1, 10, 25, 50, 75, 100\}$ during the second simulation experiment. The true extent of the object is visualized by the solid curve. The estimated center and the orientation are indicated by the yellow plus sign and the straight line, respectively. For the first instant, measurements originated from the extent and the free-space are shown by red and blue cross signs, respectively.



(a)

(b)

Figure 6.4: Results obtained during the third simulation experiment. In Fig. (a) the outputs of the tracker are illustrated at some representative time instants $k = \{1, 75, 150, 300\}$. The true extent of the object is visualized by the solid curve. The estimated center and the orientation are indicated by the yellow plus sign and the straight line, respectively. For the first instant, measurements originated from the extent and the free-space are shown by red and blue cross signs, respectively. In Fig. (b) the true and the estimated orientation are depicted.

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this thesis, we consider the challenging task of tracking dynamic objects using point cloud measurements. In recent years, there have been many attempts to construct flexible models to tackle the extended object tracking (EOT) problem. In this context, the Gaussian process-based extended object tracking (GPEOT) framework has attracted a lot of attention in the research community. This approach provides a convenient basis to recursively estimate the unknown target extent together with the kinematic state by processing point cloud data. In this work, we have contributed to the GPEOT framework by presenting several improvements for increased efficacy and efficiency. Our main contributions can be listed as follows.

- We propose an algorithm that can simultaneously track and learn latent shapes of 3D objects by processing 3D point cloud data. To this end, we describe the latent 3D extent by a radial function, which is then modeled by a GP. This formulation naturally accounts for the correlation structure within the extent, and it is highly flexible to express and learn a wide range of shapes. We establish an EOT algorithm that jointly estimates an approximate description of the GP model together with the object kinematics, including position, orientation, and velocities. Additionally, to further reduce the computational complexity, an alternative formulation is offered by utilizing plane projections. The resulting methods are demonstrated to perform successfully on simulated and real data.
- We study the Bayesian filtering problem for the GPEOT models and formulate an approximate solution based on variational Bayes techniques. The resulting algorithm effectively computes approximate posterior densities of the kinematic and the extent states. We demonstrate that the suggested algorithm

leads to improved tracking performance on simulated and real data. It is shown to be particularly robust against moderate or high levels of uncertainty included in the model.

- We suggest to utilize an alternative probabilistic model for the object extent that leads GPEOT algorithms with improved computational characteristics. In particular, a spectral-domain approximation for the underlying GP model is adopted to represent the extent. The approximation offers a basis function expansion for the GP model, where the unknown weights of the functions essentially form a parametrized description of the extent. Based on this formulation, we develop EOT algorithms for 2D and 3D problems. Through simulation experiments, we show that the resulting trackers significantly reduce the computational load and provide more satisfactory performance with a small number of extent variables.

- Existing algorithms in the GPEOT framework assume that the object of interest is star-convex. While this assumption does not introduce a severe restriction and leads to sufficiently flexible methods for many tracking applications, we focused on further extending the representation capabilities of the existing approaches. In this pursuit, we suggest to describe the unknown extent by a potential function, which is modeled by a GP. Then, a state-space model consisting of both the kinematic variables and an approximate description of the underlying GP model is developed. The resulting tracker enables tracking arbitrarily-shaped objects while learning their latent extent. We illustrate the performance of the proposed algorithm through simulation experiments.

7.0.1 Future Work

The utilization of GPs in the context of EOT leads to promising results as demonstrated in this thesis work. The proposed GPEOT algorithms are built on approximations of the original GP model, realizing inference for a finite number of inducing variables. For example, the methods making use of inducing points maintain the extent estimates at some inducing inputs, which are equidistantly located and remain fixed at their positions during the entire operation of the algorithm. In their comprehensive EOT literature review [3], Granström *et al.* brings up this issue and criticizes the method by "*..., the basis points* [utilized in the GP representation] *are uniformly distributed over the angle interval, i.e., a separation of the basis points into points for coarse and fine shape features is not possible.*" In this regard, we believe that further work can be conducted to establish a more flexible GP approximation, which can properly adjust the locations (and potentially the number) of the inducing points in an online manner. In this line of research, there will be two primary objectives: (i) to obtain a (possibly) closed form solution that adapts the locations of inducing points in an online fashion and recursively updates the posterior distribution at these points; (ii) to make use of this method in the proposed EOT algorithms to further enhance their flexibility.

Furthermore, along with precise tracking ability, the identification of the elements in the workspace is also a vital component of a perception system. Accurate label annotations substantially contribute to the robustness of an intelligent system since this information enables anticipation of future behavior and making well-informed plans accordingly. Notice that although raw LIDAR measurements are generally able to capture significant details about the underlying object surface, these impose severe problems for classification in online scenarios that are accompanied by sparsity issues due to increasing distance, change of the vantage point, and occlusions. With these in mind, the outputs of the GPEOT algorithms have been utilized for classification in 2D settings, e.g., [30, 54]. Similarly, we plan to make use of 3D extent estimates of the proposed algorithms for object classification purposes.

REFERENCES

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.

[2] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[3] K. Granstrom, M. Baum, and S. Reuter, “Extended object tracking: Introduction, overview and applications,” *J. Adv. Inf. Fusion*, vol. 12, pp. 139–174, Dec. 2017.

[4] L. E. Navarro-Serment, C. Mertz, and M. Hebert, “Pedestrian detection and tracking using three-dimensional ladar data,” *Int. J. Robot. Res.*, vol. 29, no. 12, pp. 1516–1528, 2010.

[5] A. Petrovskaya and S. Thrun, “Model based vehicle detection and tracking for autonomous urban driving,” *Autonomous Robots*, vol. 26, pp. 123–139, Apr. 2009.

[6] R. Kaestner, J. Maye, Y. Pilat, and R. Siegwart, “Generative object detection and tracking in 3D range data,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2012.

[7] H. Cho, Y.-W. Seo, B. V. Kumar, and R. R. Rajkumar, “A multi-sensor fusion system for moving object detection and tracking in urban driving environments,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014.

[8] J. W. Koch, “Bayesian approach to extended object and cluster tracking using random matrices,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, pp. 1042–1059, Jul. 2008.

[9] M. Feldmann, D. Franken, and W. Koch, “Tracking of extended objects and group targets using random matrices,” *IEEE Trans. Signal Process.*, vol. 59, pp. 1409–1420, Apr. 2011.

[10] U. Orguner, “A variational measurement update for extended target tracking with random matrices,” *IEEE Trans. Signal Process.*, vol. 60, pp. 3827–3834, Jul. 2012.

[11] J. Lan and X. R. Li, “Tracking of extended object or target group using random matrix—Part II: Irregular object,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2012.

[12] M. Baum and U. D. Hanebeck, “Random hypersurface models for extended object tracking,” in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, 2009.

[13] M. Baum and U. D. Hanebeck, “Shape tracking of extended objects and group targets with star-convex RHMs,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2011.

[14] F. Faion, M. Baum, and U. D. Hanebeck, “Tracking 3D shapes in noisy point clouds with random hypersurface models,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2012.

[15] F. Faion, A. Zea, J. Steinbring, M. Baum, and U. D. Hanebeck, “Recursive Bayesian pose and shape estimation of 3D objects using transformed plane curves,” in *Proc. IEEE ISIF Workshop Sens. Data Fusion: Trends, Solutions, Appl. (SDF)*, 2015.

[16] J. Aue, M. R. Schmid, T. Graf, J. Effertz, and P. Muehlfellner, “Object tracking from medium level stereo camera data providing detailed shape estimation using local grid maps,” in *Proc. IEEE Intell. Veh. Symp.*, 2013.

[17] M. Schütz, N. Appenrodt, J. Dickmann, and K. Dietmayer, “Multiple extended objects tracking with object-local occupancy grid maps,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2014.

[18] P. Ondruska and I. Posner, “Deep tracking: Seeing beyond seeing using recurrent neural networks,” in *Proc. AAAI Conf. Artif. Intell.*, 2016.

[19] A. Milan, S. H. Rezatofighi, A. R. Dick, I. D. Reid, and K. Schindler, “Online multi-target tracking using recurrent neural networks,” in *Proc. AAAI Conf. Artif. Intell.*, 2017.

[20] J. Dequaire, P. Ondruska, D. Rao, D. Wang, and I. Posner, "Deep tracking in the wild: End-to-end tracking using recurrent neural networks," *Int. J. Robot. Res.*, vol. 37, no. 4-5, pp. 492–512, 2018.

[21] F. Moosmann and C. Stiller, "Joint self-localization and tracking of generic objects in 3D range data," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2013.

[22] D. Held, J. Levinson, S. Thrun, and S. Savarese, "Robust real-time tracking combining 3D shape, color, and motion," *Int. J. Robot. Res.*, vol. 35, pp. 30–49, Jan. 2016.

[23] S. Kraemer, M. E. Bouzouraa, and C. Stiller, "Simultaneous tracking and shape estimation using a multi-layer laserscanner," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, 2017.

[24] D. Z. Wang, I. Posner, and P. Newman, "Model-free detection and tracking of dynamic objects with 2D lidar," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 1039–1063, 2015.

[25] E. Özkan, N. Wahlström, and S. J. Godsill, "Rao-Blackwellised particle filter for star-convex extended target tracking models," in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2016.

[26] N. Wahlström and E. Özkan, "Extended target tracking using Gaussian processes," *IEEE Trans. Signal Process.*, vol. 63, pp. 4165–4178, Aug. 2015.

[27] T. Hirscher, A. Scheel, S. Reuter, and K. Dietmayer, "Multiple extended object tracking using Gaussian processes," in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2016.

[28] M. Michaelis, P. Berthold, D. Meissner, and H.-J. Wuensche, "Heterogeneous multi-sensor fusion for extended objects in automotive scenarios using Gaussian processes and a GMPHD-filter," in *Proc. IEEE ISIF Workshop Sens. Data Fusion: Trends, Solutions, Appl. (SDF)*, 2017.

[29] K. Thormann, M. Baum, and J. Honer, "Extended target tracking using Gaussian processes with high-resolution automotive radar," in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2018.

[30] B. Tuncer, M. Kumru, E. Özkan, and A. A. Alatan, “Extended object tracking and shape classification,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2018.

[31] S. Lee and J. McBride, “Extended object tracking via positive and negative information fusion,” *IEEE Trans. Signal Process.*, vol. 67, pp. 1812–1823, Feb. 2019.

[32] M. Kumru and E. Özkan, “3D extended object tracking using recursive Gaussian processes,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2018.

[33] M. Kumru and E. Özkan, “Comments on ‘Spatio-temporal Gaussian process models for extended and group object tracking with irregular shapes’,” *arXiv preprint arXiv:2002.08065*, 2020.

[34] M. Kumru and E. Özkan, “Three-dimensional extended object tracking and shape learning using Gaussian processes,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, pp. 2795–2814, Mar. 2021.

[35] M. Kumru, H. Köksal, and E. Özkan, “Variational measurement update for extended object tracking using Gaussian processes,” *IEEE Signal Process. Lett.*, vol. 28, pp. 538–542, Feb. 2021.

[36] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013.

[37] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*. Chapman and Hall/CRC, 2004.

[38] A. H. Jazwinski, *Stochastic processes and filtering theory*. Academic Press, 1970.

[39] M. Roth and F. Gustafsson, “Computation and visualization of posterior densities in scalar nonlinear and non-Gaussian Bayesian filtering and smoothing problems,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2017.

[40] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Trans. ASME J. Basic Eng.*, vol. 82, pp. 34–45, Mar. 1960.

[41] C. M. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2006.

[42] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning.* Cambridge, MA, USA: MIT Press, 2006.

[43] N. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse Gaussian process methods: The informative vector machine," *Advances in neural information processing systems*, 2002.

[44] D. Nguyen-Tuong, J. Peters, and M. Seeger, "Local Gaussian process regression for real time online model learning," *Advances in neural information processing systems*, 2008.

[45] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[46] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *Artificial Intelligence and Statistics*, pp. 567–574, 2009.

[47] J. Quinonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *The Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[48] M. F. Huber, "Recursive Gaussian process regression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2013.

[49] M. F. Huber, "Recursive Gaussian process: On-line regression and learning," *Pattern Recogn. Lett.*, vol. 45, pp. 85–91, Aug. 2014.

[50] A. Solin and S. Särkkä, "Hilbert space methods for reduced-rank Gaussian process regression," *Statistics and Computing*, vol. 30, pp. 419–446, Aug. 2020.

[51] N. Petrov, L. Mihaylova, A. Gning, and D. Angelova, "A novel sequential Monte Carlo approach for extended object tracking based on border parameterisation," in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2011.

[52] K. Granström, P. Willett, and Y. Bar-Shalom, "An extended target tracking model with multiple random matrices and unified kinematics," in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2015.

[53] Y. Guo, Y. Li, A. Xue, R. Tharmarasa, and T. Kirubarajan, “Simultaneous tracking of a maneuvering ship and its wake using Gaussian processes,” *Signal Process.*, vol. 172, Jul. 2020.

[54] B. Tuncer, M. Kumru, and E. Özkan, “Extended target tracking and classification using neural networks,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2019.

[55] M. Michaelis, P. Berthold, T. Luettel, D. Meissner, and H.-J. Wuensche, “A merging strategy for Gaussian process extended target estimates in multi-sensor applications,” in *Proc. IEEE Intell. Veh. Symp.*, 2019.

[56] W. Martens, Y. Poffet, P. R. Soria, R. Fitch, and S. Sukkarieh, “Geometric priors for Gaussian process implicit surfaces,” *IEEE Robot. and Autom. Lett.*, vol. 2, pp. 373–380, Apr. 2017.

[57] A. O’Hagan, “Curve fitting and optimal design for prediction,” *J. R. Statist. Soc. B*, vol. 40, no. 1, pp. 1–42, 1978.

[58] G. Kurz, I. Gilitschenski, and U. D. Hanebeck, “Recursive Bayesian filtering in circular state spaces,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 31, pp. 70–87, Mar. 2016.

[59] E. Özkan, V. Šmídl, S. Saha, C. Lundquist, and F. Gustafsson, “Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters,” *Automatica*, vol. 49, pp. 1566–1575, Jun. 2013.

[60] J. L. Crassidis, F. L. Markley, and Y. Cheng, “Survey of nonlinear attitude estimation methods,” *J. Guidance Control Dyn.*, vol. 30, no. 1, pp. 12–28, 2007.

[61] F. L. Markley, “Attitude error representations for Kalman filtering,” *J. Guidance Control Dyn.*, vol. 26, no. 2, pp. 311–317, 2003.

[62] U. Maeder and M. Morari, “Attitude estimation for vehicles with partial inertial measurement,” *IEEE Trans. Veh. Tech.*, vol. 60, pp. 1496–1504, May 2011.

[63] F. L. Markley, “Attitude estimation or quaternion estimation?,” *J. Astronaut. Sci.*, vol. 52, no. 1-2, pp. 221–238, 2004.

[64] F. Gustafsson, *Statistical Sensor Fusion*. Lund, Sweden: Studentlitteratur, 2010.

[65] L. Dantanarayana, G. Dissanayake, R. Ranasinghe, and T. Furukawa, "An extended Kalman filter for localisation in occupancy grid maps," in *Proc. IEEE Int. Conf. Indust. and Inf. Syst. (ICIIS)*, 2015.

[66] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2006.

[67] A. Rivers, F. Durand, and T. Igarashi, "3D modeling with silhouettes," *ACM Trans. Graph.*, vol. 29, Jul. 2010.

[68] W. Niem, "Robust and fast modeling of 3D natural objects from multiple views," in *Proc. SPIE: Image and Video Process. II*, 1994.

[69] A. Laurentini, "How many 2D silhouettes does it take to reconstruct a 3D object?," *Comput. Vis. Image Underst.*, vol. 67, pp. 81–87, Jul. 1997.

[70] M. Baum and U. D. Hanebeck, "Extended object tracking with random hypersurface models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, pp. 149–159, Jan. 2014.

[71] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, "Blensor: Blender sensor simulation toolbox," in *Proc. Int. Symp. Visual Computing*, 2011.

[72] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, 2012.

[73] C. Lundquist, U. Orguner, and F. Gustafsson, "Extended target tracking using polynomials with applications to road-map estimation," *IEEE Trans. Signal Process.*, vol. 59, pp. 15–26, Jan. 2010.

[74] K. Granström, C. Lundquist, and U. Orguner, "Tracking rectangular and elliptical extended targets using laser measurements," in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2011.

[75] A. Zea, F. Faion, M. Baum, and U. D. Hanebeck, “Level-set random hypersurface models for tracking nonconvex extended objects,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, pp. 2990–3007, Dec. 2016.

[76] W. Cao, J. Lan, and X. R. Li, “Extended object tracking and classification using radar and ESM sensor data,” *IEEE Signal Process. Lett.*, vol. 25, pp. 90–94, Jan. 2018.

[77] H. Kaulbersch, J. Honer, and M. Baum, “A cartesian B-spline vehicle model for extended object tracking,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2018.

[78] S. Kraemer, C. Stiller, and M. E. Bouzouraa, “LiDAR-based object tracking and shape estimation using polylines and free-space information,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst. (IROS)*, 2018.

[79] J. Lan and X. R. Li, “Extended-object or group-target tracking using random matrix with nonlinear measurements,” *IEEE Trans. Signal Process.*, vol. 67, pp. 5130–5142, Oct. 2019.

[80] L. Mihaylova, A. Y. Carmi, F. Septier, A. Gning, S. K. Pang, and S. Godsill, “Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking,” *Digital Signal Process.*, vol. 25, pp. 1–16, Feb. 2014.

[81] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, “Variational inference: A review for statisticians,” *J. Amer. Statistical Assoc.*, vol. 112, no. 518, pp. 859–877, 2017.

[82] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, pp. 401–422, Mar. 2004.

[83] I. Arasaratnam and S. Haykin, “Cubature Kalman filters,” *IEEE Trans. Autom. Cont.*, vol. 54, no. 6, pp. 1254–1269, 2009.

[84] C. G. Harris, M. Stephens, *et al.*, “A combined corner and edge detector,” in *Proc. Alvey Vis. Conf.*, 1988.

[85] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[86] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2010.

[87] M. Baum, F. Faion, and U. D. Hanebeck, “Modeling the target extent with multiplicative noise,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2012.

[88] K. Granström and C. Lundquist, “On the use of multiple measurement models for extended target tracking,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2013.

[89] S. F. Kara and E. Özkan, “Multi-ellipsoidal extended target tracking using sequential Monte Carlo,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2018.

[90] B. Naujoks, P. Burger, and H.-J. Wuensche, “Fast 3D extended target tracking using NURBS surfaces,” in *Proc. Int. Trans. Syst. Conf. (ITSC)*, 2019.

[91] T. Baur, J. Böhler, S. Wirtensohn, and J. Reuter, “Tracking of spline modeled extended targets using a Gaussian mixture PHD filter,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2019.

[92] M. Baum and U. D. Hanebeck, “Extended object tracking with random hypersurface models,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, pp. 149–159, Jan. 2014.

[93] T. Baur, J. Reuter, A. Zea, and U. D. Hanebeck, “Shape estimation and tracking using spherical double Fourier series for three-dimensional range sensors,” in *Proc. Int. Conf. Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2021.

[94] M. Kok and A. Solin, “Scalable magnetic field SLAM in 3D using Gaussian process maps,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, 2018.

[95] A. Solin and M. Kok, “Know your boundaries: Constraining Gaussian processes by variational harmonic features,” in *Proc. Int. Conf. Artif. Intell. and Stat. (AISTATS)*, 2019.

[96] F. Ebert and H.-J. Wuensche, “Dynamic object tracking and 3D surface estimation using Gaussian processes and extended Kalman filter,” in *Proc. Int. Trans. Syst. Conf. (ITSC)*, 2019.

[97] S. T. O’Callaghan and F. T. Ramos, “Gaussian process occupancy maps,” *Int. J. Robot. Res.*, vol. 31, pp. 42–62, Jan. 2012.

[98] M. G. Jadidi, J. V. Miro, and G. Dissanayake, “Gaussian processes autonomous mapping and exploration for range-sensing mobile robots,” *Autonomous Robots*, vol. 42, pp. 273–290, Feb. 2018.

[99] J. Wang and B. Englot, “Fast, accurate Gaussian process occupancy maps via test-data octrees and nested Bayesian fusion,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2016.

[100] F. Ramos and L. Ott, “Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent,” *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1717–1730, 2016.

[101] S. O’Callaghan, F. T. Ramos, and H. Durrant-Whyte, “Contextual occupancy maps using Gaussian processes,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2009.

[102] R. Gill, M. W. Mueller, and R. D’Andrea, “Full-order solution to the attitude reset problem for Kalman filtering of attitudes,” *J. Guidance Control Dyn.*, pp. 1–16, Jul. 2020.

APPENDIX A

A.1 Recursive Gaussian Process Regression for Arbitrary Mean Functions

In this appendix, we will give the equations of recursive GP regression for models with arbitrary mean functions, i.e., $f(u) \sim \mathcal{GP}(\mu(u), k(u, u'))$ where there is no restriction on $\mu(\cdot)$. Based on this model, the joint distribution of a measurement, m_k, and the latent function values, $\mathbf{f}$, is described by

$$\begin{bmatrix} m_k \\ \mathbf{f} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(u_k) \\ \boldsymbol{\mu}(\mathbf{u}^\mathbf{f}) \end{bmatrix}, \begin{bmatrix} k(u_k, u_k) + R & K(u_k, \mathbf{u}^\mathbf{f}) \\ K(\mathbf{u}^\mathbf{f}, u_k) & K(\mathbf{u}^\mathbf{f}, \mathbf{u}^\mathbf{f}) \end{bmatrix}\right).$$

Then, the conditional density of m_k becomes, [42, Ch. 2.7]

$$p(m_k|\mathbf{f}) = \mathcal{N}(m_k; H_k^\mathbf{f}\mathbf{f} + c(u_k), R_k^\mathbf{f}), \tag{A.1a}$$

where

$$c(u_k) = \mu(u_k) - H_k^\mathbf{f}\boldsymbol{\mu}(\mathbf{u}^\mathbf{f}), \tag{A.1b}$$

and the definitions of $H_k^\mathbf{f}$ and $R_k^\mathbf{f}$ remain the same as given in (3.9)

$$\begin{aligned} H_k^\mathbf{f} &= H^\mathbf{f}(u_k) = K(u_k, \mathbf{u}^\mathbf{f})[K(\mathbf{u}^\mathbf{f}, \mathbf{u}^\mathbf{f})]^{-1}, \\ R_k^\mathbf{f} &= R^\mathbf{f}(u_k) = k(u_k, u_k) + R - K(u_k, \mathbf{u}^\mathbf{f})[K(\mathbf{u}^\mathbf{f}, \mathbf{u}^\mathbf{f})]^{-1}K(\mathbf{u}^\mathbf{f}, u_k). \end{aligned}$$

Similarly to the zero-mean case, (A.1) allows us to construct a state-space model, which may be regarded by a Kalman filter for recursive inference. The sole difference of the resulting model from the one given in (2.43) is that the measurement model now includes an additional term $c(u_k)$ defined in (A.1b) as

$$m_k = H_k^\mathbf{f}\,\mathbf{f}_k + c(u_k) + e_k^\mathbf{f}, \quad e_k^\mathbf{f} \sim \mathcal{N}(0, R_k^\mathbf{f}), \tag{A.2}$$

where $\mathbf{f}_k = \mathbf{f}$.

A.2 Extended Kalman Filter and Its Extensions

Some of the algorithms proposed in this thesis and some methods regarded for comparison realize inference of the state vector, x, by an extended Kalman filter (EKF) and its extensions. In this appendix, we reveal the details of these inference mechanisms.

A.2.1 Standard Extended Kalman Filtering Equations

By processing the sequentially available measurements $\mathbf{y}_{1:k}$, the filter recursively updates the estimated mean and associated covariance, which are initialized as $\hat{\mathbf{x}}_{0|-1} = \boldsymbol{\mu}_0$, $P_{0|-1} = P_0$.

A.2.1.1 Time Update

The filter computes the predicted distribution $p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1|k}, P_{k+1|k})$ in the time update step as in

$$\hat{\mathbf{x}}_{k+1|k} = F_k \hat{\mathbf{x}}_{k|k}, \tag{A.3a}$$

$$P_{k+1|k} = F_k P_{k|k} F_k^\top + Q_k. \tag{A.3b}$$

A.2.1.2 Measurement Update

The equations for the measurement update, which essentially compute the posterior distribution at time k, i.e., $p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, P_{k|k})$, are given below.

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}) \tag{A.4a}$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1} \tag{A.4b}$$

where

$$\hat{\mathbf{y}}_{k|k-1} = \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}) \tag{A.4c}$$

$$K_k = P_{k|k-1} H_k^\top S_k^{-1} \tag{A.4d}$$

$$S_k = H_k P_{k|k-1} H_k^\top + R_k \tag{A.4e}$$

$$H_k = \frac{d}{d\mathbf{x}_k} \mathbf{h}(\mathbf{x}_k)|_{\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1}} \tag{A.4f}$$

A.2.2 Extended Kalman Filtering Using Pseudo Measurements

The algorithms introduced in Ch. 3, perform inference by an EKF. However, as the pseudo-measurements, indicated by the zero vector in (A.5a), are used in the formulation of the mentioned measurement models, the implementation a dedicated EKF slightly differs from a standard EKF.

The estimate and the associated covariance are initialized as $\hat{\mathrm{x}}_{0|-1} = \boldsymbol{\mu}_0$, $P_{0|-1} = P_0$.

A.2.2.1 Measurement Update

The equations for the measurement update are as follows.

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{0} - \hat{\mathbf{y}}_{k|k-1}) \tag{A.5a}$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1} \tag{A.5b}$$

where

$$\hat{\mathbf{y}}_{k|k-1} = \mathbf{h}_k(\mathbf{y}_k, \hat{\mathbf{x}}_{k|k-1}) \tag{A.5c}$$

$$K_k = P_{k|k-1} H_k^\top S_k^{-1} \tag{A.5d}$$

$$S_k = H_k P_{k|k-1} H_k^\top + R_k \tag{A.5e}$$

$$H_k = \frac{d}{d\mathbf{x}_k} \mathbf{h}_k(\mathbf{y}_k, \mathbf{x}_k)|_{\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1}} \tag{A.5f}$$

A.2.2.2 Time Update

The time update is performed in a standard manner as in

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k} &= F_k \hat{\mathbf{x}}_{k|k}, \\ P_{k+1|k} &= F_k P_{k|k} F_k^\top + Q_k.\end{aligned}$$

A.2.3 Iterated Extended Kalman Filter Equations

In Ch. 5.6.2.1, a variant of the proposed 2D tracker is implemented by employing an iterated extended Kalman filter (IEKF). While the time update step of IEKF is identical with that of EKF, the measurement update step slightly differs. At each time instant k, IEKF performs multiple iterations for $i = 0, ..., I-1$ as in the following,

$$\hat{\mathbf{x}}_{k,i+1} = \hat{\mathbf{x}}_{k|k-1} + K_{k,i}(\mathbf{y}_k - \hat{\mathbf{y}}_{k,i}) \ , \tag{A.7a}$$

where

$$\hat{\mathbf{y}}_{k,i} = \mathbf{h}(\hat{\mathbf{x}}_{k,i}) + H_{k,i}(\hat{\mathbf{x}}_{k|k-1} - \hat{\mathbf{x}}_{k,i}) \ , \tag{A.7b}$$

$$K_{k,i} = P_{k|k-1} H_{k,i}^\top S_{k,i}^{-1} \ , \tag{A.7c}$$

$$S_{k,i} = H_{k,i} P_{k|k-1} H_{k,i}^\top + R_k \ , \tag{A.7d}$$

$$H_{k,i} = \frac{d}{d\mathbf{x}_k} \mathbf{h}(\mathbf{x}_k)|_{\mathbf{x}_k = \hat{\mathbf{x}}_{k,i}} \ , \tag{A.7e}$$

and the initial iterate is determined as

$$\hat{\mathbf{x}}_{k,0} = \hat{\mathbf{x}}_{k|k-1} \ . \tag{A.7f}$$

At the end of the iterations, the posterior mean and covariance are obtained as follows.

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k,I} \tag{A.8a}$$

$$P_{k|k} = P_{k|k-1} - K_{k,I} H_{k,I} P_{k|k-1} \tag{A.8b}$$

A.3 Update of the Reference Quaternion

As presented in Ch. 3.5.1.2, the orientation of the 3D object is described by a reference quaternion, $\mathbf{q}_{\text{ref}}$, and a deviation vector, $\mathbf{a}$. For maintaining this description, we follow the standard approach provided within the framework of MEKF, e.g., [60, 61, 62].

In particular, suppose that at time k, we have the reference quaternion $\mathbf{q}_{\text{ref},k}$, and $\hat{\mathbf{a}}_{k|k}$ represents the estimated deviation vector, which is computed by the measurement update of the filter. We first update the reference quaternion in accordance with (3.28),

$$\mathbf{q}_{\text{ref},k+1} = \delta\mathbf{q}(\hat{\mathbf{a}}_{k|k}) \odot \mathbf{q}_{\text{ref},k}, \tag{A.9}$$

where $\odot$ denotes the quaternion product defined in (3.26). Subsequently, the deviation vector is reset to zero as

$$\hat{\mathbf{a}}_{k|k} = \mathbf{0}. \tag{A.10}$$

Please note that the covariance of the deviation vector is kept unchanged, as suggested in [61]. This approach is referred to as the *zero-order approximation*, [102], and it enables effective estimation of the orientation, as demonstrated by the comprehensive performance analysis. For a detailed investigation of the reference orientation update mechanisms, interested readers can refer to [102] and the references therein.

A.4 Details of the Matrices Used in the Rotational Motion Model

In Ch. 3.5.1.2, the state-space model for describing the dynamics of the rotational motion is given by (3.36). In this appendix, we reveal the appendix of the matrices used in the model.

$$\begin{aligned} F_k^r &= \exp(A_k^r T) \\ &= \begin{bmatrix} \exp(\frac{T}{2}[-\hat{\boldsymbol{\omega}}_{k|k}\times]) & T\exp(\frac{T}{2}[-\hat{\boldsymbol{\omega}}_{k|k}\times]) \\ 0_3 & I_3 \end{bmatrix} \end{aligned} \tag{A.11a}$$

where

$$\begin{aligned} \exp(\frac{T}{2}[-\hat{\boldsymbol{\omega}}_{k|k}\times]) = I_3 &+ \frac{\sin(\frac{T}{2}|\hat{\boldsymbol{\omega}}_{k|k}|)}{|\hat{\boldsymbol{\omega}}_{k|k}|}[-\hat{\boldsymbol{\omega}}_{k|k}\times] \\ &+ \frac{1-\cos(\frac{T}{2}|\hat{\boldsymbol{\omega}}_{k|k}|)}{|\hat{\boldsymbol{\omega}}_{k|k}|^2}[-\hat{\boldsymbol{\omega}}_{k|k}\times]^2 \end{aligned} \tag{A.11b}$$

$|\cdot|$ indicates the Euclidean norm.

$$\begin{aligned}G_k &= \left(\int_0^T \exp(A_k^r \tau) d\tau\right) B \\ &= \begin{bmatrix} \int_0^T \exp(\frac{\tau}{2}[-\hat{\boldsymbol{\omega}}_{k|k}\times])d\tau & \int_0^T \tau \exp(\frac{\tau}{2}[-\hat{\boldsymbol{\omega}}_{k|k}\times])d\tau \\ 0_3 & \int_0^T I_3 d\tau \end{bmatrix} B \end{aligned} \tag{A.12a}$$

where

$$\begin{aligned}\int_0^T \exp(\frac{\tau}{2}[-\hat{\boldsymbol{\omega}}_{k|k}\times])d\tau = TI_3 &+ \frac{2(1-\cos(\frac{T}{2}|\hat{\boldsymbol{\omega}}_{k|k}|))}{|\hat{\boldsymbol{\omega}}_{k|k}|^2}[-\hat{\boldsymbol{\omega}}_{k|k}\times] \\ &+ \frac{T - \frac{2}{|\hat{\boldsymbol{\omega}}_{k|k}|}\sin(\frac{T}{2}|\hat{\boldsymbol{\omega}}_{k|k}|)}{|\hat{\boldsymbol{\omega}}_{k|k}|^2}[-\hat{\boldsymbol{\omega}}_{k|k}\times]^2 \end{aligned} \tag{A.12b}$$

$$\begin{aligned}&\int_0^T \tau \exp(\frac{\tau}{2}[-\hat{\boldsymbol{\omega}}_{k|k}\times])d\tau = \frac{T^2}{2} I_3 \\ &+ \frac{1}{|\hat{\boldsymbol{\omega}}_{k|k}|^2}\left(\frac{4}{|\hat{\boldsymbol{\omega}}_{k|k}|}\sin(\frac{T}{2}|\hat{\boldsymbol{\omega}}_{k|k}|) - 2T\cos(\frac{T}{2}|\hat{\boldsymbol{\omega}}_{k|k}|)\right)[-\hat{\boldsymbol{\omega}}_{k|k}\times] \\ &+ \frac{1}{|\hat{\boldsymbol{\omega}}_{k|k}|^2}\left(\frac{T^2}{2} + \frac{2T}{|\hat{\boldsymbol{\omega}}_{k|k}|}\sin(\frac{T}{2}|\hat{\boldsymbol{\omega}}_{k|k}|)\right. \\ &\left. + \frac{4}{|\hat{\boldsymbol{\omega}}_{k|k}|^2}(\cos(\frac{T}{2}|\hat{\boldsymbol{\omega}}_{k|k}|) - 1)\right)[-\hat{\boldsymbol{\omega}}_{k|k}\times]^2 \end{aligned} \tag{A.12c}$$

$$\int_0^T I_3 d\tau = TI_3 \tag{A.12d}$$

A.5 The Unscented Transformation

In Chapter 4.2.1 of this thesis, the following intractable expectations are approximately computed by the unscented transformation as

$$\begin{aligned}\mathbb{E}_{q_x^{(i+1)}}[\underbrace{\tilde{H}(\bar{\mathbf{x}}_k)^\top R_k^{-1}\tilde{H}(\bar{\mathbf{x}}_k)}_{g_1(\bar{\mathbf{x}}_k)}] &\approx \sum_{j=0}^M \pi^j g_1(\bar{\mathbf{x}}_k^j) \\ \mathbb{E}_{q_x^{(i+1)}}[\underbrace{\tilde{H}(\bar{\mathbf{x}}_k)^\top R_k^{-1}(\mathbf{y}_k - \mathbf{c}_k)}_{g_2(\bar{\mathbf{x}}_k)}] &\approx \sum_{j=0}^M \pi^j g_2(\bar{\mathbf{x}}_k^j),\end{aligned}$$

where $\{\bar{\mathbf{x}}_k^j\}_{j=0}^M$ and $\{\pi^j\}_{j=0}^M$ denote the sigma-points and the associated weights, and $q_x^{(i+1)}(\bar{\mathbf{x}}_k)$ is the variational iterate for the kinematic state defined as $q_x^{(i+1)}(\bar{\mathbf{x}}_k) = \mathcal{N}(\boldsymbol{\mu}_x^{(i+1)}, P_x^{(i+1)})$.

In the literature, there are various methods to select the sigma-points and their weights. In this study, we follow the standard approach in [30] and specify

$$\bar{\mathbf{x}}_k^j \triangleq \begin{cases} \boldsymbol{\mu}_x^{(i+1)}, & j = 1 \\ \boldsymbol{\mu}_x^{(i+1)} + \left[\sqrt{\frac{m}{1-\bar{\pi}} P_x^{(i+1)}}\right]_{:,j-1}, & 2 \leq j \leq m+1 \\ \boldsymbol{\mu}_x^{(i+1)} - \left[\sqrt{\frac{m}{1-\bar{\pi}} P_x^{(i+1)}}\right]_{:,j-m-1}, & m+2 \leq j \leq 2m+1 \end{cases},$$

$$\pi^j \triangleq \begin{cases} \bar{\pi}, & j = 1 \\ \frac{1-\bar{\pi}}{2m}, & 2 \leq j \leq 2m+1 \end{cases},$$

where m is dimension of the kinematic state vector, $[\cdot]_{:,j}$ denotes the j^{th} column of the matrix argument, $\sqrt{\cdot}$ indicates the matrix square root operation, and we choose $\bar{\pi} = \frac{1}{2m+1}$.

A.6 Measurement Model of the Inducing Point-based GPEOT

In the existing GPEOT approaches, which rely on an inducing point approximation for the underlying GP model, a single point measurement $\mathbf{y}_{k,l}$ is formulated as [26]

$$\mathbf{y}_{k,l} = \mathbf{h}_l(\mathbf{x}_k) + \mathbf{e}_{k,l}, \ \mathbf{e}_{k,l} \sim \mathcal{N}(0, R_{k,l}), \tag{A.13}$$

where

$$\mathbf{h}_l(\mathbf{x}_k) = \mathbf{c}_k + \mathbf{p}_l(\mathbf{c}_k) H_{k,l}^{\mathbf{f}} \mathbf{f}_k \,, \tag{A.14a}$$

$$H_{k,l}^{\mathbf{f}} = H^{\mathbf{f}}\left(\theta_{k,l}^L\right) = k(\theta_{k,l}^L, \boldsymbol{\theta}^{\mathbf{f}})\, K(\boldsymbol{\theta}^{\mathbf{f}}, \boldsymbol{\theta}^{\mathbf{f}})^{-1}, \tag{A.14b}$$

$$R_{k,l} = \mathbf{p}_l(\mathbf{c}_k) R_{k,l}^{\mathbf{f}} \mathbf{p}_l(\mathbf{c}_k)^\top + R \,, \tag{A.14c}$$

$$R_{k,l}^{\mathbf{f}} = R^{\mathbf{f}}\left(\theta_{k,l}^L\right) = k(\theta_{k,l}^L, \theta_{k,l}^L) - H_{k,l}^{\mathbf{f}}\, k(\boldsymbol{\theta}^{\mathbf{f}}, \theta_{k,l}^L) \,, \tag{A.14d}$$

$$\mathbf{p}_l(\mathbf{c}_k) = \left[\cos(\theta_{k,l}^{\mathrm{G}}) \ \sin(\theta_{k,l}^{\mathrm{G}})\right]^\top , \tag{A.14e}$$

$$\theta_{k,l}^L = \theta_{k,l}^{\mathrm{G}} - \psi_k \,, \tag{A.14f}$$

$$\theta_{k,l}^G = \angle(\mathbf{y}_{k,l} - \mathbf{c}_k) \,, \tag{A.14g}$$

(A.14h)

where $\boldsymbol{\theta}^{\mathbf{f}} \triangleq \left[\theta_1^{\mathbf{f}} \ldots \theta_m^{\mathbf{f}}\right]^\top$ denotes the inducing vector consisting of the input angles. The formulation is based on a GP model for the radial function, i.e., $f(\theta) \sim \mathcal{GP}(0, k(\theta, \theta'))$. The GP model is approximately expressed for the function values $\mathbf{f} \triangleq \left[f(\theta_1^{\mathbf{f}}) \ldots f(\theta_m^{\mathbf{f}})\right]^\top$ at the inducing inputs. The unified state vector $\mathbf{x}_k$ includes both the kinematic and the extent state, i.e., $\mathbf{x}_k = \begin{bmatrix}\bar{\mathbf{x}}_k^\top & \mathbf{f}_k^\top\end{bmatrix}^\top$. The i-j^{th} entry of the covariance matrix is $K(\boldsymbol{\theta}^{\mathbf{f}}, \boldsymbol{\theta}^{\mathbf{f}})_{ij} = k(\theta_i^{\mathbf{f}}, \theta_j^{\mathbf{f}})$. The definitions of $H^{\mathbf{f}}(\cdot)$ and $R^{\mathbf{f}}(\cdot)$ follow directly from the standard GP regression formulae.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Kumru, Murat

EDUCATION

Degree	Institution	Year of Graduation
M.S.	Middle East Technical University	2015
B.S.	Middle East Technical University	2012

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2021-Present	Kuartis Inc.	Lead Engineer
2014-2021	Middle East Technical University	Research/ Teaching Assistant
2012-2014	Aselsan Inc.	Engineer

PUBLICATIONS

1. M. Kumru, E. Özkan, "Three-Dimensional Extended Object Tracking and Shape Learning Using Gaussian Processes," IEEE Transactions on Aerospace and Electronic Systems, vol. 57, pp. 2795–2814, Mar. 2021.

2. M. Kumru, H. Köksal, E. Özkan, "Variational Measurement Update for Extended Object Tracking Using Gaussian Processes," IEEE Signal Processing Letters, vol. 28, pp. 538–542, Feb. 2021.

International Conference Publications

1. B. Tuncer, M. Kumru, E. Özkan, “Extended Target Tracking and Classification Using Neural Networks,” International Conference on Information Fusion, 2019.

2. M. Kumru, E. Özkan, “3D Extended Object Tracking Using Recursive Gaussian Processes,” International Conference on Information Fusion, 2018.

3. B. Tuncer, M. Kumru, E. Özkan, A. A. Alatan, “Extended Object Tracking and Shape Classification,” International Conference on Information Fusion, 2018.

4. İ. K. Erünsal, K. Ahiska, M. Kumru, K. Leblebicioğlu. “An Approach for System Identification in Unmanned Surface Vehicles,” IEEE International Conference on Control, Automation and Systems, 2017.

5. M. Kumru, E. G. Debada, L. Makarem, D. Gillet, “Mobility-on-demand Scenarios Relying on Lightweight Autonomous and Connected Vehicles for Large Pedestrian Areas and Intermodal Hubs,” IEEE International Conference on Intelligent Transportation Engineering, 2017.

6. M. Kumru, K. Leblebicioğlu, İ.K. Erünsal, K. Ahiska, “A Survey on Tactical Control Algorithms for Path Tracking Unmanned Surface Vehicles,” IEEE International Conference on Control, Automation, Robotics and Vision, 2016.