



TECHNICAL UNIVERSITY OF MUNICH

DEPARTMENT OF INFORMATICS

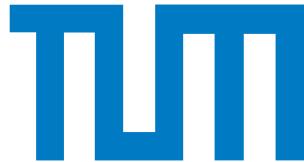
Master's Thesis in Robotics, Cognition, Intelligence

**Track-Before-Detect Smoothing for  
Multiple Extended Objects using Random  
Finite Sets**

Boqian Yu







TECHNICAL UNIVERSITY OF MUNICH

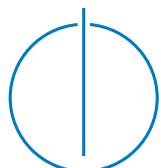
DEPARTMENT OF INFORMATICS

Master's Thesis in Robotics, Cognition, Intelligence

## **Track-Before-Detect Smoothing for Multiple Extended Objects using Random Finite Sets**

## **Track-Before-Detect Smoothing für Mehrere Ausgedehnte Objekte mit Random Finite Sets**

Author:	Boqian Yu
Supervisor:	Prof. Dr.-Ing. Matthias Althoff
Advisor:	Egon Ye, M.Sc.
Submission Date:	26.06.2020



I confirm that this master's thesis is my own work and I have documented all sources and material used.

Ich versichere, dass ich diese Master's Thesis selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Munich, 26.06.2020

Boqian Yu

## Acknowledgments

Upon the completion of this thesis, I am grateful to all those who have offered me invaluable encouragement and support during the whole course of my master study, without whom I could not have achieved so much in the past two years.

First of all, the profound gratitude should go to my supervisor Egon Ye at BMW. He has offered me countless helpful ideas and instructions regarding mathematics, algorithms, programming and paper writing in the discussions throughout my internship and master thesis at BMW Autonomous Driving Campus, and has also enlightened my consideration of the career path. He has always been a role model for me in both research and work aspects.

Secondly, I would like to express my heartfelt thanks to all the professors and teachers who have taught and helped me at TUM. Their inspirational and conscientious teaching has provided me with a solid basis for the composing of this thesis and will always be of great value to my future academic research. I am particularly grateful to Professor Matthias Althoff for his AI and CPS lectures as well as his kind suggestions on the paper, which have aroused my interest in the current research field and deepened my understanding of academic research.

I also want to thank all my colleges at BMW and fellow students at TUM, with whom the discussions and coffee breaks have always been enjoyable and meaningful. Learning about different cultures and ideas has never been so much of a pleasure.

Finally, I am indebted to my family for supporting me to implement my own ideas and to pursue my goals in life. Their guidance and suggestions are of incomparable value to me.



# Abstract

Aiming at improving both safety and comfort in transportation, the development of Automated Driving Systems (ADSs) has fundamentally changed the conventional control of vehicle. As researches and applications advance rapidly forth, growing demand on the reference data to validate the functionalities of the ADS quickly exceeds the capability of conventional methods based on the Global Navigation Satellite System (GNSS), real-time kinematics and hand labeling in both quantity and quality. To handle this foreseeable shortage of ground-truth data, novel schemes and methodologies have been investigated in the literature of the recent years addressing automatic ground-truth generation.

This thesis proposes a new method to generate accurate tracking results that can be used as reference data for validating the functionalities of the ADS. A forward-backward smoother for multiple extended objects is applied based on the Labeled Multi-Bernoulli (LMB) Random Finite Set (RFS) and the recursive Gaussian process, under the assumption that object contours can be modeled as star-convex shapes. A robust birth policy with a backward filter is further proposed to solve the conflict between robustness and completeness of tracking. Therein, cluster candidates are evaluated based on a quality measure to only initialize objects from more reliable clusters in the forward pass. Missing states will then be recovered by the backward filter through post-processing the unassociated data after the smoothing process. Simulations and real-world experiments demonstrate superior performance of the proposed method in both cardinality and individual state estimation compared to the naive LMB filter and smoother for extended objects.



# Contents

<b>Acknowledgments</b>	iii
<b>Abstract</b>	v
<b>1 Introduction</b>	1
1.1 Motivation . . . . .	1
1.2 Multiple Object Tracking . . . . .	3
1.3 Objective and Outline . . . . .	5
<b>2 Fundamentals</b>	7
2.1 Notations . . . . .	7
2.2 Motion Models . . . . .	7
2.2.1 The CV Model . . . . .	8
2.2.2 The CTRV Model . . . . .	9
2.2.3 The CTRA Model . . . . .	10
2.3 Finite Set Statistics . . . . .	11
2.3.1 Definition . . . . .	12
2.3.2 Common Classes of RFSs . . . . .	13
2.3.3 Bayesian Multi-Object Inference . . . . .	15
2.4 Extended Object Modeling . . . . .	16
2.4.1 Star-convex Shape . . . . .	17
2.4.2 Gaussian Process . . . . .	17
2.4.3 Extended State Vector . . . . .	19
<b>3 Forward-Backward Labeled Multi-Bernoulli Smoothing</b>	23
3.1 Observation Set Partitioning . . . . .	24
3.1.1 K-means . . . . .	24
3.1.2 Distance Partitioning . . . . .	25
3.1.3 Prediction Partitioning . . . . .	25
3.1.4 Comparison of Partitioning Methods . . . . .	25
3.2 Forward Filtering . . . . .	27
3.2.1 Prediction Step . . . . .	27
3.2.2 Robust Birth Policy . . . . .	28
3.2.3 Multiple Extended Object Likelihood . . . . .	32
3.2.4 Update Step . . . . .	32
3.2.5 State Extraction . . . . .	36
3.3 Backward Smoothing . . . . .	37

3.4	Track-Before-Detect State Generation . . . . .	38
<b>4</b>	<b>Performance Evaluation</b>	<b>43</b>
4.1	Test Perspectives and Metrics . . . . .	43
4.2	Simulations . . . . .	44
4.2.1	Simulation Setup . . . . .	45
4.2.2	Single Object Simulations . . . . .	45
4.2.3	Multiple Object Simulation . . . . .	51
4.3	Real-world Experiments . . . . .	52
4.3.1	Data Acquisition . . . . .	54
4.3.2	Test Results . . . . .	55
4.3.3	Influence of Detection Probability . . . . .	56
<b>5</b>	<b>Conclusion and Perspective</b>	<b>61</b>
5.1	Conclusion . . . . .	61
5.2	Perspective . . . . .	62
<b>List of Figures</b>		<b>65</b>
<b>List of Tables</b>		<b>67</b>
<b>Bibliography</b>		<b>69</b>

# 1 Introduction

## 1.1 Motivation

"Automated driving is the greatest mobility revolution since the invention of automobile" [1], as can often be read from the media and industrial reports. Aiming at improving both safety and comfort in transportation, the Automated Driving Systems (ADSs) are fundamentally changing the conventional control of vehicle by gradually excluding the unobservable and uncontrollable human driver from the closed "Human-Vehicle-Road" controlling cycle, and have gained increasing attention in many relevant research and industry fields. A typical architecture of the ADS comprises three processing layers as in Fig. 1.1: the perception layer acquires data from various types of sensors or from network connections; the decision making layer interprets information from the data and determines the appropriate behaviors accordingly; the execution layer carries out the desired behaviors. These three layers cooperate with each other, exchange data at different levels, and are all crucial to the realization of various automated driving functionalities.

The past decade has witnessed rapid advancement in the transition process towards completely driverless driving as pointed out in Fig. 1.2, according to which the driving automation is divided into six levels. Above the third level of conditional driving automation, the driving system should be capable of performing the driving tasks without the permanent presence of the human driver, referring mainly to the timely and



Figure 1.1: Three processing layers of a typical automated vehicle (adapted from [2]).

## 1 Introduction

---

Level	Name	Narrative definition	DDT		DDT fallback	ODD
			Sustained lateral and longitudinal vehicle motion control	OEDR		
<b>Driver performs part or all of the DDT</b>						
0	No Driving Automation	The performance by the driver of the entire DDT, even when enhanced by active safety systems.	Driver	Driver	Driver	n/a
1	Driver Assistance	The sustained and ODD-specific execution by a driving automation system of either the lateral or the longitudinal vehicle motion control subtask of the DDT (but not both simultaneously) with the expectation that the driver performs the remainder of the DDT.	Driver and System	Driver	Driver	Limited
2	Partial Driving Automation	The sustained and ODD-specific execution by a driving automation system of both the lateral and longitudinal vehicle motion control subtasks of the DDT with the expectation that the driver completes the OEDR subtask and supervises the driving automation system.	System	Driver	Driver	Limited
<b>ADS ("System") performs the entire DDT (while engaged)</b>						
3	Conditional Driving Automation	The sustained and ODD-specific performance by an ADS of the entire DDT with the expectation that the DDT fallback-ready user is receptive to ADS-issued requests to intervene, as well as to DDT performance-relevant system failures in other vehicle systems, and will respond appropriately.	System	System	Fallback-ready user (becomes the driver during fallback)	Limited
4	High Driving Automation	The sustained and ODD-specific performance by an ADS of the entire DDT and DDT fallback without any expectation that a user will respond to a request to intervene.	System	System	System	Limited
5	Full Driving Automation	The sustained and unconditional (i.e., not ODD-specific) performance by an ADS of the entire DDT and DDT fallback without any expectation that a user will respond to a request to intervene.	System	System	System	Unlimited

Figure 1.2: SAE levels of automation for automated driving [3]. DDT denotes *Dynamic Driving Task*, OEDR denotes *Object and Event Detection and Response*, ODD denotes *Operational Design Domain*, ADS denotes *Automated Driving System*.

proper response to the behaviors of the surrounding objects and events, which requires higher autonomy of the vehicle especially in environment understanding based on the sensor data, i.e., the second architect layer mentioned above. For such an objective, much effort has been devoted from both the academia and the industry to the development and improvement of data interpretation algorithms, including object detection, object tracking, motion prediction etc. Large amount of high-quality ground-truth data is required to validate and evaluate of these algorithms [4], for which the demand quickly exceeds the capability of conventional methods based on GNSS and real-time kinematics in both quantity and quality [5].

To cope with the foreseeable shortage of ground-truth data, novel methodologies such as evaluation without ground-truth [6, 7] and automatic ground-truth generation [8, 9] have been investigated in recent years, among which a promising idea is to generate reference data directly from the sensor recordings. By post-processing the recordings and incorporating the future data to the estimation of each time step with smoothing algorithms, more accurate reference data can be generated compared to online processing methods. The generated data can then be utilized as reference data while testing the functions of automated driving systems.

Inspired by this idea, a new scheme is proposed in this thesis to generate more reliable reference data that can be used as ground-truth: based on LiDAR data, an offline forward-backward Bayesian tracking is performed for multiple extended objects, which benefits from simultaneous estimation of kinematics and object contours and information fusion of past, current, and future sensor recordings; furthermore, tract-before-detect estimations can also be generated for object states, for which only few measurements are available at the beginning, in a "look-back" process from the future to solve the conflict between robustness and completeness for tracking, and thus provides more reliable reference data.

## 1.2 Multiple Object Tracking

The objective of multi-object tracking is to jointly estimate the amount of existing tracks as well as their individual states from a sequence of noisy and cluttered recordings. Apart from temporary occlusion, the substantial difference between multi-object and single-object tracking is the varying amount of tracks due to object births and deaths [10, 11]. Algorithms for multi-object tracking can be classified into three paradigms according to their solution to the *data association problem* [12]: Joint Probabilistic Data Association methods [10, 13] perform weighted state updates using all possible measurement-to-object association hypotheses based on the association probabilities and propagate the approximated posterior distributions; Multiple Hypothesis Tracking methods [14, 15] preserve all association hypotheses and incorporate the subsequent measurements to solve the association uncertainty problem; and the Random Finite Set (RFS) framework [11] involves modeling and propagating the multiple object states at each time step as a whole multi-object state.

The fundamental methodology of the RFS framework is to model the collection of individual object states, including object births, deaths, occlusion and clutter, as finite set-valued random variables (e.g., in Fig. 1.3), and cast the data association problem in the classic Bayesian inference framework using Finite Set Statistics (FISST) developed in [11]. Tracking via propagating the posterior densities using the RFS framework has gained popularity in recent years, of which the Probability Hypothesis Density filter [16, 17], the Multi-Target Multi-Bernoulli filter [11, 18], the Generalized Labeled Multi-Bernoulli (GLMB) filter [19, 20] and its first moment approximation LMB filter [21] have been widely applied [22, 23, 24]. The formulae of the corresponding RFS-based

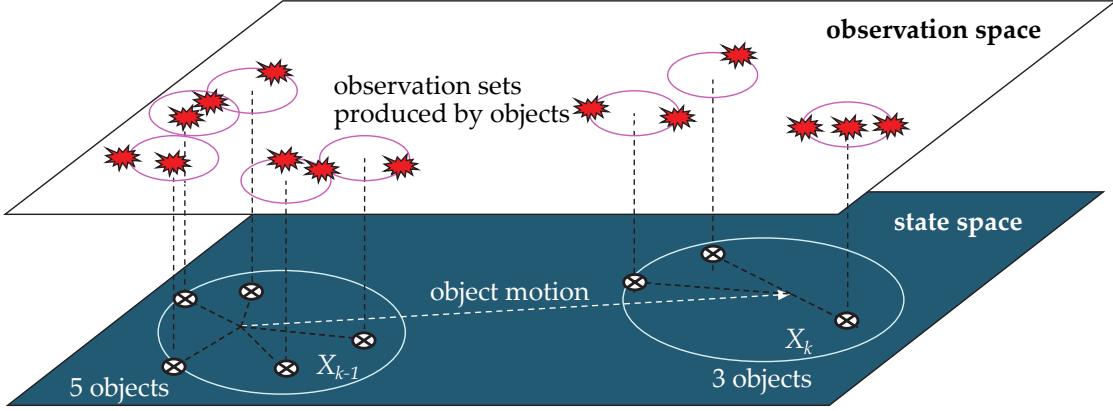


Figure 1.3: Basic idea of the RFS framework, including alteration in object amount, clutter and multiple measurement points from the same object [29].

smoothers have also been developed in the literature [25, 26, 27, 28].

The previously-mentioned approaches model objects as single points without extents; however, one is often additionally interested in estimating the shape of objects. Objects that can occupy multiple resolution cells of the sensor are denoted as extended objects, for which trackers have been developed by integrating extended object models. In some simple cases, object contours can be modeled as basic shapes like circles [30, 31] or rectangles [32, 33]. The random matrix model is frequently applied to estimate elliptical shapes [34, 35, 36], while the random hypersurface model proposed in [37] is flexible to model various kinds of contours such as elliptical or star-convex [38, 39]. The recursive Gaussian process proposed in [40, 41, 42] can efficiently propagate shapes approximated by certain basis vectors [43, 44].

In most object tracking scenarios, measurement extraction involves some thresholding operation to prevent false tracks, which could in turn result in loss of information when the signal-to-noise ratio is low, as the example illustrated in Fig. 1.4. A higher threshold usually leads to more robust estimations yet less complete tracks, while a low threshold normally yields unreliable estimations due to inadequate information for initialization.

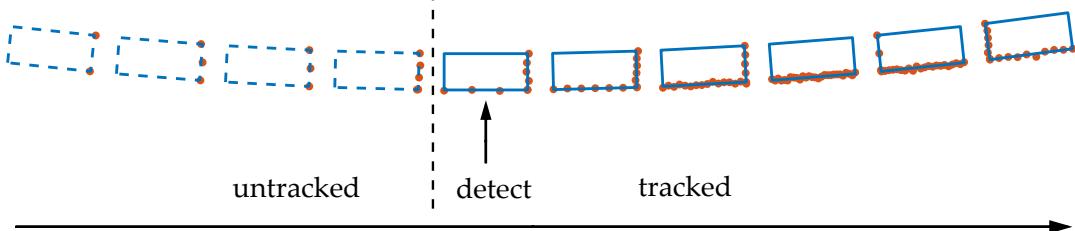


Figure 1.4: Example of track-before-detect states in a sequence of point cloud data, where solid and dashed lines denote tracked and untracked states.

The term *track-before-detect* refers to tracking approaches that process all sensor data without applying a threshold [45], and is typically used for object tracking on radar intensity images [46, 47, 48]. This term can be generalized to describe approaches based on LiDAR data [49], where the number of measurements decreases as the object gets further from the sensor.

### 1.3 Objective and Outline

The main objective of this thesis is to provide a novel approach of generating reference data for multiple extended objects to serve the evaluation of automated driving systems. To achieve this objective, a practical framework of processing point cloud data sequence recorded by a LiDAR sensor shall be investigated. The data association problem shall be solved effectively, while typical behaviors in multi-object tracking like object birth/death and occlusion shall be properly modeled. To solve the conflict of completeness and robustness in multi-object tracking and generate reliable estimation for both trajectory and object contour, the prerequisite for object initialization shall be controlled, while missing states shall be possibly recovered afterwards. In modeling extended objects, fewer constraints on contour shape shall be assumed so that objects of various types and geometries could be tracked in the same scenario simultaneously. For evaluation of the approach, a prototype shall be implemented to illustrate the performance in estimating both individual object states (kinematics and contour) and the multi-object cardinality. Finally, the proposed approach shall be validated in experiments with both simulational and real-world datasets through comparison against the reference results.

The rest of this thesis is organized as follows: Chapter 2 provides fundamental theoretical background for the subsequent chapters, including a brief review of motion models considered in this thesis, the basics of FISST to model multi-object states, and the recursive Gaussian process for modeling and propagating extended objects. Combined together, these theories form the basic components of the proposed approach. Chapter 3 then presents the theories of the proposed tracking scheme in this thesis. The detailed workflow of generating track-before-detect estimations from LiDAR measurement sequence is described in individual steps, namely the forward filtering process, the backward smoothing process and the backward filtering process. Besides, a robust birth policy for object initialization is discussed to allow object birthing only when enough information is available. The corresponding measure to recover missing states in the early steps is then discussed in the backward filtering process. Evaluation results are shown in chapter 4 to test the performance of the approach, including both simulations and real-world experiments. In the simulations, the scenario setup is first specified, followed by test scenarios for both single object and multi-object tracking. In the real-world experiments, the sensor platform for data acquisition is introduced, followed by tests to evaluate the generated tracks by comparing to the recordings of the Differential Global Positioning System (DGPS). Finally in chapter 5, conclusions of the explorations in this thesis are given, and the future work of the proposed approach is outlooked.



## 2 Fundamentals

### 2.1 Notations

In this thesis, different forms of symbols are used to demonstrate variables of different types: scalars are denoted with regular small letters (e.g.,  $x$ ), vectors with underlined small letters (e.g.,  $\underline{x}$ ), constant scalars, sets or matrices with regular capital letters (e.g.,  $N, X, Z, F$ ), and spaces with blackboard bold letters (e.g.,  $\mathbb{X}, \mathbb{L}$ ). Bold face letters (e.g.,  $\mathbf{x}, \mathbf{X}$ ) are used for labeled variables (introduced in section 2.3.2) to distinguish from the unlabeled ones.

The multi-object exponential for some real-valued function  $h$  on a finite set  $X$  is denoted by

$$h^X \triangleq \prod_{\underline{x} \in X} h(\underline{x}), \quad (2.1)$$

with  $h^\emptyset = 1$  by convention. The inner product of functions is denoted by

$$\langle f, g \rangle \triangleq \int f(x)g(x)dx. \quad (2.2)$$

The inclusion function is given by

$$1_Y(X) \triangleq \begin{cases} 1, & \text{if } X \subseteq Y, \\ 0, & \text{otherwise.} \end{cases} \quad (2.3)$$

If  $X = \{x\}$ , the notation  $1_Y(x)$  is used instead of  $1_Y(\{x\})$  for brevity.

### 2.2 Motion Models

A motion model is a dynamic model used to describe the kinematic transition of object states over time. Various motion models exist for both linear and nonlinear state transitions [50, 51], the choice of which depends on the desired type of state transition for the specific application. A common mathematical demonstration of a motion model is given in state space as:

$$\underline{x}_k = f_k(\underline{x}_{k-1}) + \underline{q}_k, \quad (2.4)$$

where  $f_k$  is the transition function of the state,  $\underline{q}_k \sim \mathcal{N}(\underline{0}, Q_k)$  is the process noise vector assumed to be a Wiener Process increment (white Gaussian noise) with covariance matrix  $Q_k$ . This section gives a brief review of three motion models on the two dimensional

plane that are used in this thesis, namely the Constant Velocity (CV) model, the Constant Turn Rate Velocity (CTRV) model and the Constant Turn Rate Acceleration (CTRA) model, among which the CV model is a linear model and the CTRV and CTRA models are nonlinear. As the foundation of their application to the Bayesian inference processes in the succeeding sections, this section also discusses the linearization of these models in the discret state space regarding their insertion into the Kalman filter [52].

### 2.2.1 The CV Model

A CV motion model is able to describe a straight motion at a constant velocity, and describes a state by its position and velocity, i.e.,

$$\underline{x} = [x \quad y \quad \dot{x} \quad \dot{y}]^\top. \quad (2.5)$$

For a CV model, the kinematics along each axis is independent, and the process noise term  $q_k$  in (2.4) represents the disturbance caused by some white noise velocity changes. The transition over time follows from:

$$\underline{x}_{k+1} = \underbrace{\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{F_k} \underline{x}_k + G \begin{bmatrix} w_{v,x} \\ w_{v,y} \end{bmatrix}, \quad (2.6)$$

where  $w_{v,x} \sim \mathcal{N}(0, \sigma_{v,x}^2)$ ,  $w_{v,y} \sim \mathcal{N}(0, \sigma_{v,y}^2)$  are the white Gaussian noise terms for velocities in the x- and y-directions, and

$$G = \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \quad (2.7)$$

is the noise gain matrix. Assuming the velocity noise to be equally distributed in x- and y-directions (i.e.,  $\sigma_{v,x}^2 = \sigma_{v,y}^2 = \sigma_v^2$ ), the covariance matrix  $Q$  of process noise can be formulated as:

$$Q = G\sigma_v^2 G^\top = \sigma_v^2 \begin{bmatrix} \frac{1}{4}\Delta t^4 & 0 & \frac{1}{2}\Delta t^3 & 0 \\ 0 & \frac{1}{4}\Delta t^4 & 0 & \frac{1}{2}\Delta t^3 \\ \frac{1}{2}\Delta t^3 & 0 & \Delta t^2 & 0 \\ 0 & \frac{1}{2}\Delta t^3 & 0 & \Delta t^2 \end{bmatrix}. \quad (2.8)$$

Since the CV model is linear, a Kalman filter can be easily applied by directly using the matrix  $F_k$  above as the transition matrix to perform tracking for objects whose motion is constrained on a straight line.

### 2.2.2 The CTRV Model

Despite the simplicity of the CV model, it is usually not practical in real-world applications since the kinematics of most objects are correlated among axes. Common counter-examples are typical road objects like pedestrians, cyclists, vehicles etc. with some certain heading orientations, whose motion is mostly constrained to the direction of their heading, as illustrated in Fig. 2.1.

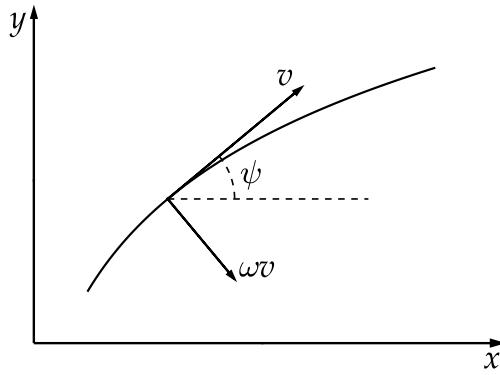


Figure 2.1: The CTRV motion model

To better model the motion of these oriented objects, the CTRV model explicitly considers a constant yaw rate  $\omega$  for the change of the yaw angle  $\psi$  beside the position and constant velocity of an oriented object, i.e.,

$$\underline{x} = [x \ y \ \psi \ v \ \omega]^\top. \quad (2.9)$$

The corresponding state equation in the continuous state space is

$$\dot{\underline{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \cos(\psi) \\ v \sin(\psi) \\ \omega \\ 0 \\ 0 \end{bmatrix} + \dot{\underline{q}}, \quad (2.10)$$

where  $\dot{\underline{q}}$  is the white Gaussian process noise vector. Under the concept of direct discretization [53], the exact solution of this nonlinear differential equation in the discrete state space can be obtained by first ignoring  $\dot{\underline{q}}$  and solving the homogeneous differential equation. After inserting the piecewise constant noise vector directly into the discret

state system, the transition equation of the CTRV motion model is given as follows:

$$\begin{aligned}\underline{x}_{k+1} &= \underline{x}_k + \int_{k\Delta t}^{(k+1)\Delta t} \dot{\underline{x}}(\tau) d\tau \\ &= \underline{x}_k + \underbrace{\left[ \begin{array}{c} \frac{2v_k}{\omega_k} \sin\left(\frac{\omega_k \Delta t}{2}\right) \cos\left(\psi_k + \frac{\omega_k \Delta t}{2}\right) \\ \frac{2v_k}{\omega_k} \sin\left(\frac{\omega_k \Delta t}{2}\right) \sin\left(\psi_k + \frac{\omega_k \Delta t}{2}\right) \\ \omega_k \Delta t \\ 0 \\ 0 \end{array} \right]}_{f(\underline{x}_k)} + G \begin{bmatrix} w_v \\ w_\omega \end{bmatrix},\end{aligned}\quad (2.11)$$

where  $f(\underline{x}_k)$  is the transition function,  $w_v \sim \mathcal{N}(0, \sigma_v^2)$  and  $w_\omega \sim \mathcal{N}(0, \sigma_\omega^2)$  are the white Gaussian noise for velocity and yaw rate, and the noise gain matrix approximated by the corresponding columns of the Jacobian matrix of  $f(\underline{x}_k)$  w.r.t.  $\underline{x}_k$  due to its nonlinearity:

$$G = \left[ \frac{\partial f}{\partial v} \Big|_{v_k} \quad \frac{\partial f}{\partial \omega} \Big|_{\omega_k} \right] = \left[ \begin{array}{cc} \frac{2}{\omega_k} \sin\left(\frac{\omega_k \Delta t}{2}\right) CW & -\frac{v_k}{\omega_k^2} [SW + \omega_k \Delta t CW - \sin \psi_k] \\ \frac{2}{\omega_k} \sin\left(\frac{\omega_k \Delta t}{2}\right) SW & \frac{v_k}{\omega_k^2} [CW + \omega_k \Delta t SW - \sin \psi_k] \\ 0 & \Delta t \\ 1 & 0 \\ 0 & 1 \end{array} \right], \quad (2.12)$$

with  $SW = \sin\left(\psi_k + \frac{\omega_k \Delta t}{2}\right)$ ,  $CW = \cos\left(\psi_k + \frac{\omega_k \Delta t}{2}\right)$ . The covariance matrix is then given by:

$$Q = G \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix} G^\top. \quad (2.13)$$

Since the CTRV motion model is nonlinear, an Extended Kalman Filter (EKF) is utilized when applying Bayesian state estimation over time. The transition function in (2.11) is linearized by using the first-order Taylor expansion as an approximation in the filtering process. Note that in the process above, the continuous system is first discretized and then linearized (known as linearized discretization) for smaller loss of accuracy in approximation [54]. Another alternative is to first apply linearization to the continuous system followed by discretization (known as discretized linearization), which will yield slightly different results [55].

### 2.2.3 The CTRA Model

As the generalization and expansion of the CTRV model, the CTRA motion model further adopts the acceleration in the heading direction as a state variable (Fig. 2.2) to account for the motion of oriented objects based on the disturbances in yaw rate and acceleration rather than in yaw rate and velocity , i.e.,

$$\underline{x} = [x \ y \ \psi \ v \ \omega \ a]^\top. \quad (2.14)$$

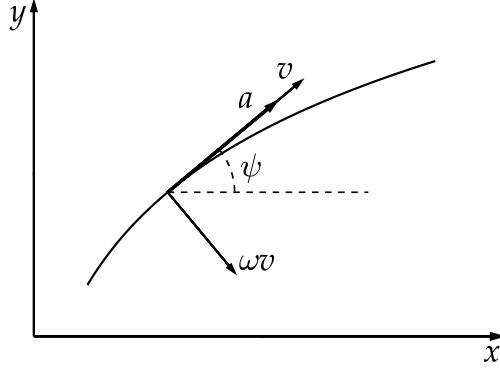


Figure 2.2: The CTRA motion model

and the continuous state space differential equation is given by

$$\dot{\underline{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{v} \\ \dot{\omega} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} v \cos(\psi) \\ v \sin(\psi) \\ \omega \\ a \\ 0 \\ 0 \end{bmatrix} + \dot{\underline{q}}, \quad (2.15)$$

where  $\dot{\underline{q}}$  is the white Gaussian process noise vector. Similar to (2.11), the discrete transition function for the CTRA model can be formulated as:

$$\underline{x}_{k+1} = \underline{x}_k + \int_{k\Delta t}^{(k+1)\Delta t} \dot{\underline{x}}(\tau) d\tau = f(\underline{x}_k) + G \begin{bmatrix} w_\omega \\ w_a \end{bmatrix}, \quad (2.16)$$

with  $w_\omega \sim \mathcal{N}(0, \sigma_\omega^2)$ ,  $w_a \sim \mathcal{N}(0, \sigma_a^2)$  as the white Gaussian noise for acceleration and yaw rate, and the noise gain and covariance matrices can be approximated by:

$$G = \begin{bmatrix} \frac{\partial f}{\partial \omega} \Big|_{\omega_k} & \frac{\partial f}{\partial a} \Big|_{a_k} \end{bmatrix}, \quad Q = G \begin{bmatrix} \sigma_\omega^2 & 0 \\ 0 & \sigma_a^2 \end{bmatrix} G^\top. \quad (2.17)$$

According to the state description and the noise term, the CTRA model is especially favorable for modeling vehicles [56], as the disturbances of human operation on steering and accelerating/braking are directly modeled by the two noise parameters  $\sigma_\omega^2$  and  $\sigma_a^2$ , which can be easily adjusted according to certain applications. Similar to the CTRV model, an EKF is required for object tracking with the CTRA model due to nonlinearity.

## 2.3 Finite Set Statistics

As reviewed in section 1.2, a generalization of the single-object Bayesian tracking framework to multi-object situations is feasible by modeling the individual object states and the measurements as whole multi-object state variables using Random Finite

Sets (RFSs), respectively. The Finite Set Statistics (FISST) [11] facilitates an intuitive application of the random finite set theory to multi-object tracking applications by casting the problem into the classic framework of Bayesian inference [16]. Consequently, the multi-object Bayes filter is a rigorous extension of the standard single-object Bayes filter to multi-object tracking: an RFS naturally represents the uncertainty on the number of objects in a multi-object state and uses random vectors to represent the states of the individual objects. Furthermore, the RFS can be conveniently used to describe the measurement process which returns a random number of measurements whose values are represented by random vectors.

This section briefly summarizes the main concepts of the FISST and introduces the multi-object Bayes filter. The tutorial papers [57, 58] are suitable for the first steps into the FISST, the book [11] provides a more detailed mathematical foundation for the FISST and RFS-based multi-object tracking, and the thesis [12] presents comprehensive discussions on the application of the FISST from a multi-object filtering perspective.

### 2.3.1 Definition

A *Random Finite Set* (RFS) is simply a random variable that take values as (unordered) finite sets, i.e., a finite-set-valued random variable. The essential difference between an RFS and a random vector is that: for an RFS, the number of constituent points is random and the points themselves are also random, distinct and unordered; whereas for a random vector there is exactly one constituent point which is random. An RFS  $X$  on the state space  $\mathbb{X} \subseteq \mathbb{R}^d$  (where  $d$  is the dimension of the state vectors) can be completely specified by a discrete distribution that characterizes the cardinality (number of points), and a family of symmetric joint distributions that characterizes the distribution of the points conditional on the cardinality [59]. Let  $\mathbb{N}$  and  $\mathbb{N}^+$  denote the non-negative integers and positive integers respectively, a cardinality distribution  $\rho(\cdot)$  on  $\mathbb{N}$  determines the total number of points in the set, and for any nonzero cardinality  $n \in \mathbb{N}^+$ , a probability distribution  $P_n(\cdot)$  on the product space  $\mathbb{X}^n = \mathbb{X} \times \dots \times \mathbb{X}$  determines the joint distribution of the points given that there are  $n$  points in total, i.e.,

$$X = \begin{cases} \emptyset & , \text{if } n = 0, \\ \{\underline{x}^{(1)}, \dots, \underline{x}^{(n)}\}, \underline{x}^{(i)} \in \mathbb{X} & , \text{if } n \in \mathbb{N}^+. \end{cases} \quad (2.18)$$

Intuitively, to simulate an RFS, first sample a non-negative integer  $n$  from  $\rho(\cdot)$ , then draw a sample from  $P_n(\cdot)$  to get  $n$  points of  $\mathbb{X}$ . For example, in a Poisson RFS, the cardinality is Poisson distributed with a given mean, and the points are independently and identically distributed according to a given distribution on  $\mathbb{X}$  (see the next section for more details on the common classes of RFSs).

Analogous to random vectors, a probability density function  $\pi(X)$  facilitates the representation of the uncertainty about an RFS  $X$ , which incorporates the uncertainty about the number of objects and their individual states. The multi-object probability

density depends on the number of objects contained in a certain RFS realization  $X$  and is given by

$$\pi(X) = \begin{cases} \pi(\emptyset) & , \text{ if } X = \emptyset, \\ \pi\left(\{\underline{x}^{(1)}\}\right) & , \text{ if } X = \{\underline{x}^{(1)}\}, \\ \pi\left(\{\underline{x}^{(1)}, \underline{x}^{(2)}\}\right) & , \text{ if } X = \{\underline{x}^{(1)}, \underline{x}^{(2)}\}, \\ \vdots & \vdots \end{cases} \quad (2.19)$$

Due to the dependency of (2.19) on the number of objects, the evaluation of the integral over the probability density function of an RFS requires the utilization of a set integral defined in [11]:

$$\int f(X)\delta X = \sum_{i=0}^{\infty} \frac{1}{i!} \int_{\mathbb{X}^i} f\left(\{\underline{x}^{(1)}, \dots, \underline{x}^{(i)}\}\right) d\underline{x}^{(1)} \cdots d\underline{x}^{(i)}. \quad (2.20)$$

The probability density function of an RFS always satisfies  $\int \pi(X)\delta X = 1$  by definition. These concepts enable the extension of classic Bayesian inference framework to multi-object scenarios, which will be discussed in the subsequent sections.

### 2.3.2 Common Classes of RFSs

As can be concluded from the previous section, various types of RFSs mainly differ in the probability distribution over cardinality and/or individual state vector. This subsection introduces several common classes of RFSs that are encountered later in the subsequent sections of this thesis, namely the Poisson RFS, the multi-Bernoulli RFS, the labeled multi-Bernoulli RFS and the generalized labeled multi-Bernoulli RFS.

#### Poisson RFS

A Poisson RFS  $X$  on  $\mathbb{X}$  is uniquely defined by its intensity function  $v$ . The Poisson RFSs have the unique property that the distribution of the cardinality of  $X$  is Poisson with mean  $\lambda = \int v(\underline{x})d\underline{x}$ , i.e.,

$$\rho(n) = e^{-\lambda} \cdot \frac{\lambda^n}{n!}, \quad (2.21)$$

and for a given cardinality the elements of  $X$  are each independently and identically distributed with probability density  $v(\cdot)/\lambda$ . The intensity function, also known in the literature as the Probability Hypothesis Density (PHD), completely characterizes a Poisson RFS. Consequently, the density function of a Poisson RFS is given by

$$\pi(X) = e^{-\lambda} v^X \quad (2.22)$$

In tracking applications, a Poisson RFS with intensity function  $\kappa(z) = \lambda_c \cdot c(z)$  is commonly used to model the false alarm (clutter) process with an expected mean

number of  $\lambda_c$  false alarms and probability density  $c(z)$  (usually assumed as a uniform distribution over the entire observation space due to the absence of clutter information).

### Multi-Bernoulli RFS

An intuitive way to represent the uncertainty about the existence of an object is the usage of a Bernoulli RFS. A Bernoulli RFS  $X$  is a singleton with probability  $r$  and is empty with probability  $1 - r$ , where  $r$  is commonly referred to as the existence probability of the singleton that represents a certain state with spatial distribution  $p(x)$  on the state space  $\mathbb{X}$ . A Bernoulli RFS is completely described by the parameter pair  $(r, p)$ , with which the probability density is given by

$$\pi(X) = \begin{cases} 1 - r & , \text{ if } X = \emptyset, \\ r \cdot p(\underline{x}) & , \text{ if } X = \{\underline{x}\}. \end{cases} \quad (2.23)$$

A multi-Bernoulli RFS  $X$  is the extension of the Bernoulli RFS to the multi-object case. Under the assumption that the Bernoulli RFSs  $X^{(i)}$  of the objects are independent of each other, the multi-Bernoulli RFS  $X$  for  $M$  objects is given by  $X = \bigcup_{i=1}^M X^{(i)}$ . Thus, the parameter set  $\{(r^{(i)}, p^{(i)})\}_{i=1}^M$  completely defines a multi-Bernoulli RFS, with which the probability density function is given by

$$\pi\left(\{\underline{x}^{(1)}, \dots, \underline{x}^{(n)}\}\right) = \prod_{j=1}^M (1 - r^{(j)}) \sum_{1 \leq i_1 \neq \dots \neq i_n \leq M} \prod_{j=1}^n \frac{r^{(i_j)} p^{(i_j)}(\underline{x}^{(j)})}{1 - r^{(i_j)}}, \quad (2.24)$$

specially,  $\pi(\emptyset) = \prod_{i=1}^M (1 - r^{(i)})$  corresponds to the probability that none of the  $M$  objects exists. The mean cardinality of a multi-Bernoulli RFS is  $\hat{N} = \sum_{i=1}^M r^{(i)}$  and can be used to estimate the expected number of objects that are currently present.

### Labeled Multi-Bernoulli RFS

In order to simplify the estimation of state trajectories, a state  $x \in \mathbb{X}$  can be augmented by a label  $\ell \in \mathbb{L}$  to mark the state series of the same object, i.e.,  $x^{(i)} = (\underline{x}^{(i)}, \ell^{(i)})$ . A labeled RFS  $X = \{x^{(i)}\}_{i=1}^M$  defined on the state space  $\mathbb{X}$  and discrete label space  $\mathbb{L}$  has only distinct labels within each realization, i.e., let  $\mathcal{L}(X) = \{\ell : (\underline{x}, \ell) \in X\}$  denote the set of labels in  $X$ , the distinct label indicator

$$\Delta(X) = \begin{cases} 1, & \text{if } |\mathcal{L}(X)| = |X|, \\ 0, & \text{if } |\mathcal{L}(X)| \neq |X| \end{cases} \quad (2.25)$$

always satisfies  $\Delta(X) = 1$ . A Labeled Multi-Bernoulli (LMB) RFS is a multi-Bernoulli RFS with distinct labels attached to non-empty Bernoulli components, defined by the parameter set

$$\boldsymbol{\pi} = \left\{ (r^{(\ell)}, p^{(\ell)}) \right\}_{\ell \in \mathbb{L}}, \quad (2.26)$$

where  $r^{(\ell)}$  is the existence probability and  $p^{(\ell)}$  the probability density of the component with label  $\ell \in \mathbb{L}$ . The density of an LMB RFS is then given by [19]

$$\pi(\underline{x}) = \Delta(\underline{x}) w(\mathcal{L}(\underline{x})) p^{\underline{x}}, \quad (2.27)$$

with

$$\begin{aligned} p(\underline{x}, \ell) &= p^{(\ell)}(\underline{x}), \\ w(\mathcal{L}(\underline{x})) &= \prod_{i \in \mathbb{L}} \left(1 - r^{(i)}\right) \prod_{\ell \in \mathcal{L}(\underline{x})} \frac{1_{\mathbb{L}}(\ell) r^{(\ell)}}{1 - r^{(\ell)}}. \end{aligned} \quad (2.28)$$

### Generalized Labeled Multi-Bernoulli RFS

In an LMB RFS, all Bernoulli components are assumed to be statistically independent, and the weight (2.28) of each realization  $\underline{x}$  depends on the existence probabilities  $r^{(\ell)}$  of the individual tracks  $\ell \in \mathcal{L}(\underline{x})$ . Consequently, the cardinality distribution of an LMB RFS follows the one of the multi-Bernoulli distribution, which is unimodal. The Generalized Labeled Multi-Bernoulli (GLMB) RFS [19] is a generalization of the LMB RFS which facilitates arbitrary weights and cardinality distributions. The density function of a GLMB RFS defined on the state space  $\mathbb{X}$  and the label space  $\mathbb{L}$  is given by

$$\pi(\underline{x}) = \Delta(\underline{x}) \sum_{\xi \in \Xi} w^{(\xi)}(\mathcal{L}(\underline{x})) \left[ p^{(\xi)} \right]^{\underline{x}}, \quad (2.29)$$

where  $\Xi$  is a discrete hypothesis index set that assembles multiple realizations (combinations) from a set of track labels. The weights  $w^{(\xi)}(L)$  and the spatial distributions  $p^{(\xi)}$  satisfy the normalization conditions

$$\begin{aligned} \sum_{L \subseteq \mathbb{L}} \sum_{\xi \in \Xi} w^{(\xi)}(L) &= 1, \\ \int_{\underline{x} \in \mathbb{X}} p^{(\xi)}(\underline{x}, \ell) d\underline{x} &= 1. \end{aligned} \quad (2.30)$$

A GLMB RFS can be interpreted as a mixture of multi-object realizations (or hypotheses). Each term  $\xi$  in the mixture (2.29) consists of a weight  $w^{(\xi)}(\mathcal{L}(\underline{x}))$  that only depends on the labels of this multi-object state realization, and a multi-object exponential  $\left[ p^{(\xi)} \right]^{\underline{x}}$  that depends on the probability distributions of all constituent components in it. While the components of a GLMB RFS are not statistically independent, an LMB RFS comprises only statistically independent tracks. Also, it can be observed that an LMB is a special case of a GLMB (with one single term in  $\Xi$ ).

#### 2.3.3 Bayesian Multi-Object Inference

In the classic single-object Bayesian inference framework [60], the hidden state  $\underline{x}_k \in \mathbb{X}$  at time  $k$  is partially observed as  $\underline{z}_k \in \mathbb{Z}$  in the observation space according to the likelihood function  $g_k(\underline{z}_k | \underline{x}_k)$ , and the stochastic dynamics of the system over time is described

by some initial prior  $p_0(\underline{x})$  and a sequence of conditional probability distributions  $\underline{x}_{k+1} \sim f_{k+1|k}(\underline{x}_{k+1}|\underline{x}_k)$ , called the dynamic model or the transition model, under the Markov assumption. The prediction of the state distribution to the next time step can be obtained by:

$$p_{k+1|k}(\underline{x}) = \langle f_{k+1|k}(\cdot|\cdot), p_{k|k} \rangle. \quad (2.31)$$

Incorporated with the measurement  $\underline{z}_{k+1}$ , all information about the state up to time step  $k + 1$  is encapsulated in the posterior distribution, given by the Bayes rule:

$$p_{k+1|k+1}(\underline{x}) = \frac{g_k(\underline{z}_{k+1}|\underline{x})p_{k+1|k}(\underline{x})}{\langle g(\underline{z}_{k+1}|\cdot), p_{k+1|k} \rangle}. \quad (2.32)$$

Together, the equations (2.31) and (2.32) compose the forward pass (or the filtering process) of Bayesian inference. Furthermore, given the measurements up to a time step  $T$ , the marginal posterior distribution of the state  $\underline{x}_k, k < T$  conditioned on these measurements can be calculated in the backward pass (or the smoothing process) as:

$$p_{k|T}(\underline{x}) = p_{k|k}(\underline{x}) \left\langle f_{k+1|k}(\cdot|\underline{x}), \frac{p_{k+1|T}}{p_{k+1|k}} \right\rangle. \quad (2.33)$$

By replacing the random state vectors with RFSs and applying the FISST discussed in section 2.3.1, the classic Bayesian inference framework can be extended to satisfy the demand of multi-object state estimation, yielding equations in similar forms [11]:

$$\pi_{k+1|k}(X) = \langle f_{k+1|k}(X|\cdot), \pi_{k|k} \rangle, \quad (2.34)$$

$$\pi_{k+1|k+1}(X) = \frac{g_k(Z_{k+1}|X)\pi_{k+1|k}(X)}{\langle g(Z_{k+1}|\cdot), \pi_{k+1|k} \rangle}, \quad (2.35)$$

$$\pi_{k|T}(X) = \pi_{k|k}(X) \left\langle f_{k+1|k}(\cdot|X), \frac{\pi_{k+1|T}}{\pi_{k+1|k}} \right\rangle, \quad (2.36)$$

among which the Chapman-Kolmogorov equation is applied to account for the multi-object prediction in (2.34), and  $X, Z$  are the RFSs of the multi-object states and measurements, respectively.

## 2.4 Extended Object Modeling

Among various modeling approaches for extended objects reviewed in section 1.2, this thesis utilizes the two dimensional star-convex model to demonstrate the extent of an object due to its flexibility in modeling irregular shapes and simplicity in parameterization. This section presents the fundamental theories required for applying this model to the estimation of object extents. Note that in this section, all discussions are based on single-object scenarios, i.e., no data association problem is considered.

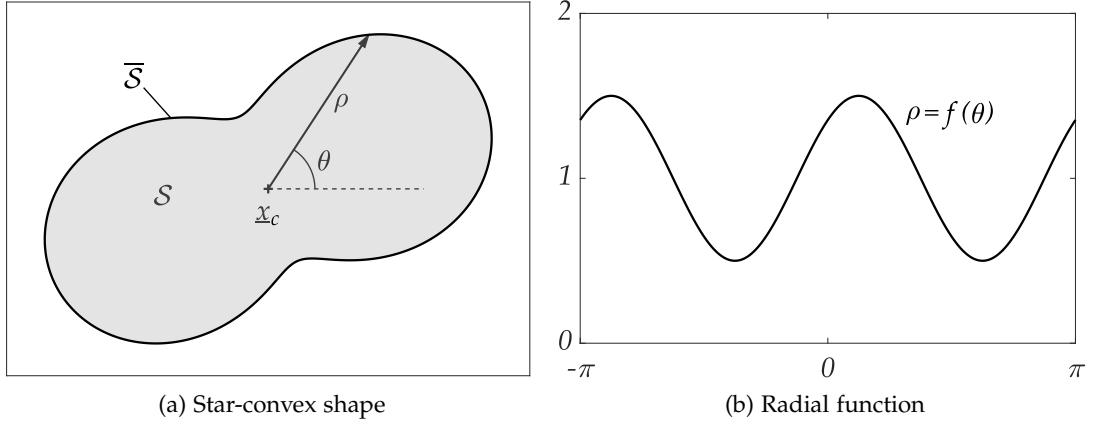


Figure 2.3: Example of a star-convex shape and the corresponding radial function

#### 2.4.1 Star-convex Shape

A closed shape  $\mathcal{S} \subset \mathbb{R}^2$  is star-convex, if there exists one reference point  $\underline{x}_c$  within  $\mathcal{S}$  such that for every  $\underline{x} \in \mathcal{S}$ , the line segment between  $\underline{x}_c$  and  $\underline{x}$  is fully contained in  $\mathcal{S}$  [38], as exemplified in Fig. 2.3a. With some known reference point, the contour of a star-convex shape can be fully described by a radial function  $\rho = f(\theta)$  in the local polar coordinate system that maps angles to radii, i.e.,

$$\bar{\mathcal{S}} = \left\{ \underline{x}_c + \underline{p}(\theta)f(\theta), \theta \in (-\pi, \pi] \right\}, \quad (2.37)$$

where  $\bar{\mathcal{S}}$  denotes the contour of  $\mathcal{S}$  and  $\underline{p}(\theta) \triangleq [\cos(\theta) \ \sin(\theta)]^\top$  is the orientation vector for the angle  $\theta$ . Compared to other models like rectangles or ellipses, a star-convex model is less restrictive on the geometric characteristics of the shape, and can provide flexible modeling for most regular and irregular shapes at any desired level of approximation given appropriate parameterization of the radial function [61], which will be discussed in the following sections.

#### 2.4.2 Gaussian Process

A Gaussian Process (GP) [62]

$$f(u) \sim \mathcal{GP} (\mu(u), k(u, u')) \quad (2.38)$$

is a stochastic process uniquely defined by its mean  $\mu(u)$  and covariance (kernel) function  $k(u, u')$ . A GP can be considered as a distribution over functions whose values follow the multivariate Gaussian distribution

$$\begin{bmatrix} f(u_1) \\ \vdots \\ f(u_n) \end{bmatrix} \sim \mathcal{N} \left( \underline{\mu}, K \right), \quad (2.39)$$

where

$$\underline{\mu} = \begin{bmatrix} \mu(u_1) \\ \vdots \\ \mu(u_n) \end{bmatrix}, K = \begin{bmatrix} k(u_1, u_1) & \dots & k(u_1, u_n) \\ \vdots & \ddots & \vdots \\ k(u_n, u_1) & \dots & k(u_n, u_n) \end{bmatrix} \quad (2.40)$$

are the mean vector and covariance matrix, respectively.

Given the measurements  $\underline{z} = [z_1 \dots z_N]^\top$  and their corresponding input  $\underline{u} = [u_1 \dots u_N]^\top$ , a GP can be used to learn function values  $\underline{f} = [f_1 \dots f_{N_f}]^\top$  at certain points  $\underline{u}^f = [u_1^f \dots u_{N_f}^f]^\top$ . Assuming the mean function  $\underline{\mu} = \underline{0}$ , the joint distribution of  $\underline{z}$  and  $\underline{f}$  is given by

$$\begin{bmatrix} \underline{z} \\ \underline{f} \end{bmatrix} \sim \mathcal{N}\left(\underline{0}, \begin{bmatrix} K(\underline{u}, \underline{u}) & K(\underline{u}, \underline{u}^f) \\ K(\underline{u}^f, \underline{u}) & K(\underline{u}^f, \underline{u}^f) \end{bmatrix}\right), \quad (2.41)$$

which yields

$$p(\underline{f}|\underline{z}) = \mathcal{N}\left(\underline{m}^f, P^f\right), \quad (2.42)$$

with

$$\begin{aligned} \underline{m}^f &= K(\underline{u}^f, \underline{u}) K^{-1}(\underline{u}, \underline{u}) \underline{z}, \\ P^f &= K(\underline{u}^f, \underline{u}^f) - K(\underline{u}^f, \underline{u}) K^{-1}(\underline{u}, \underline{u}) K(\underline{u}, \underline{u}^f). \end{aligned} \quad (2.43)$$

The type of functions that can be modeled by a GP strongly depends on the choice of the kernel function, which should be selected according to the characteristics of the function in the specific applications [62]. The exponential sine square kernel [63] is applied in this thesis to model the periodic radial function of star-convex shapes with period  $2\pi$ :

$$k(u, u') = \sigma_F^2 \cdot e^{-\frac{2 \sin^2\left(\frac{|u-u'|}{2}\right)}{l^2}} + \sigma_R^2. \quad (2.44)$$

As shown in Fig. 2.4, the constant parameters  $l, \sigma_F^2, \sigma_R^2$  in (2.44) together decide the form of the kernel function. The parameter  $l$  denotes the length scale which can be

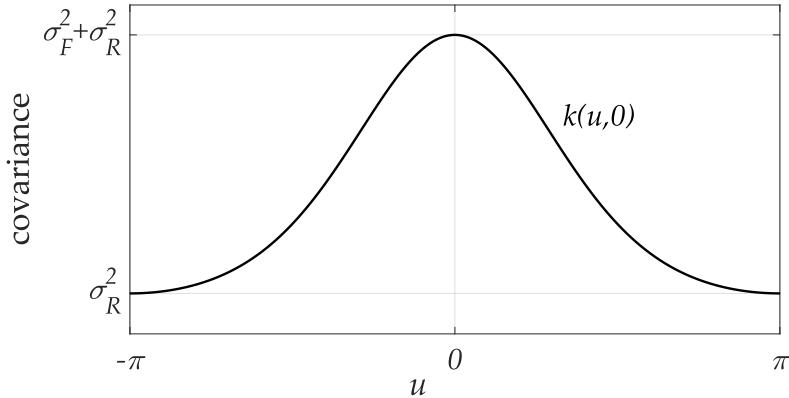


Figure 2.4: The exponential sine square kernel function with period  $2\pi$

interpreted as the distance in the inputs that is necessary for a significant change in the outputs, or simply the influence range of an input point. The variance  $\sigma_F^2$  is the prior variance of the signal amplitude and  $\sigma_R^2$  is the variance contribution from the mean function  $\mu(u)$ . Selecting  $\sigma_F^2$  and  $\sigma_R^2$  to be small allows to keep the estimated function values close to the initial prior  $f_0$ , which can result in a robust estimation in scenarios with high clutter yet strong prior knowledge [44].

### 2.4.3 Extended State Vector

As the star-convex shape model is utilized in this thesis to demonstrate the contour of an extended object, its radial function needs to be efficiently estimated and propagated over time with a sequence of data, i.e., we are interested in the posterior  $p(f|Z_{1:k})$ . The direct application of (2.41) using all previous measurements is in general not tractable, since the time complexity increase cubically with the growth of the measurement number  $N$  due to the matrix inverse operation in (2.43). Instead, the recursive Gaussian process proposed in [40] uses a finite set of function values (the *basis vector*)  $\underline{\rho}^b = [\rho_1^b, \dots, \rho_{N^b}^b]^\top$  at specific *basis points*  $\underline{\theta}^b = [\theta_1^b, \dots, \theta_{N^b}^b]^\top$  to approximate the function, where  $N^b \ll N$ . The principle of this method can be understood as to estimate the lengths of several radial lines at certain angles to "support" the contour, and can provide approximation of  $f$  at different levels of accuracy depending on  $N^b$  and other parameters [43], as shown in the example of Fig. 2.5. An alternative to basis vectors is to use the first  $N^f$  Fourier coefficients that define a Fourier series expansion [38], which can be estimated in the same manner.

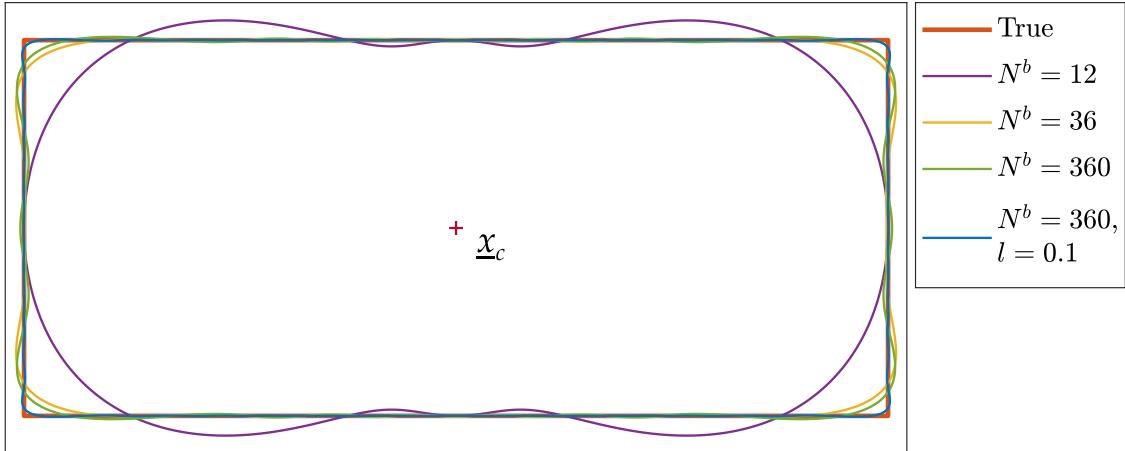


Figure 2.5: Example of contour approximation with Gaussian process at different levels of accuracy, controlled by the number of basis points  $N^b$  and the length scale  $l$  in Gaussian process regression. Basis points for all approximations are equidistantly spaced in range  $(-\pi, \pi]$ . The first three approximations use length scale  $l = 0.4$ , while the last one uses  $l = 0.1$ .

In the recursive Gaussian process with basis vector  $\underline{\rho}^b$ , we have for the posterior at time  $k$  under the Bayesian inference framework:

$$p(\underline{\rho}^b | Z_{1:k}) \propto p(Z_k | \underline{\rho}^b, Z_{1:k-1}) \cdot p(\underline{\rho}^b | Z_{1:k-1}), \quad (2.45)$$

If we assume  $Z_k$  to be conditionally independent of the previous measurements  $Z_{1:k-1}$ , then  $\underline{\rho}^b$  itself can already function as sufficient statistic of  $Z_{1:k-1}$  and yields:

$$p(Z_k | \underline{\rho}^b, Z_{1:k-1}) = p(Z_k | \underline{\rho}^b). \quad (2.46)$$

Let  $\underline{\theta}_k, \underline{\rho}_k$  denote the measurements  $Z_k = \{z_{k,1}, \dots, z_{k,N_k}\}$  expressed in the local polar coordinate system w.r.t. the reference point, the assumption above would be exact if  $\{\underline{\theta}_i\}_{i=1}^{k-1}$  is a subset of the basis points  $\underline{\theta}^b$ , i.e.,  $\underline{\rho}^b$  contains all information in the measurements of the past. Also, a good approximation would be achieved if  $N^b$  and  $\underline{\theta}^b$  are chosen carefully according to the typical length scale of the polar angles from measurements, such that these angles are close to those in  $\underline{\theta}^b$ . Consequently, the basis vector  $\underline{\rho}^b$  can be recursively estimated with a Kalman filter. Assuming zero mean function for the Gaussian process, the propagation of the posterior follows the conventional *Predict-Update* scheme [40]:

$$\begin{aligned} J_k &= K(\underline{\theta}_k, \underline{\theta}^b) \cdot K(\underline{\theta}^b, \underline{\theta}^b)^{-1}, \\ \underline{\rho}_{k|k-1} &= J_k \cdot \underline{\rho}_{k-1}^b, \\ C_{k|k-1} &= K(\underline{\theta}_k, \underline{\theta}_k) + J_k \cdot (C_{k-1}^b - K(\underline{\theta}^b, \underline{\theta}^b)) \cdot J_k^\top, \\ G_k &= C_{k-1}^b \cdot J_k^\top \cdot (C_{k|k-1} + \sigma_M^2 I)^{-1}, \\ \underline{\rho}_k^b &= \underline{\rho}_{k-1}^b + G_k \cdot (\underline{\rho}_k - \underline{\rho}_{k|k-1}), \\ C_k^b &= (I - G_k \cdot J_k) \cdot C_{k-1}^b, \end{aligned} \quad (2.47)$$

where  $\underline{\rho}_{k|k-1} \in \mathbb{R}^{N_k}$  and  $C_{k|k-1} \in \mathbb{R}^{N_k} \times \mathbb{R}^{N_k}$  are the predicted radial function values at  $\underline{\theta}_k$ ,  $G_k$  is the Kalman gain matrix,  $\sigma_M^2$  is the observation noise variance and  $I$  is the identity matrix of proper dimensions. The initial mean and covariance of the basis vector are  $\underline{\rho}_0^b = \underline{0}$  and  $C_0^b = K(\underline{\theta}^b, \underline{\theta}^b)$ . Equations (2.47) allow simultaneous propagation of both the kinematic state and the contour parameters of an extended object. Combining the kinematic state vector  $\underline{x}^k$  (e.g.,  $\underline{x}^k = [x \ y \ \psi \ v \ \omega \ a]^\top$  for the CTRA motion model) with the basis vector, an individual extended object can be fully described by its extended state vector:

$$\underline{x} = \begin{bmatrix} \underline{x}^k \\ \underline{\rho}^b \end{bmatrix} \in \mathbb{R}^{|\underline{x}^k| + N^b}. \quad (2.48)$$

The corresponding transition matrix and the process noise matrix of the extended state vector are given by:

$$F = \begin{bmatrix} F^k & 0 \\ 0 & e^{-\tau \Delta t} I \end{bmatrix}, \quad Q = \begin{bmatrix} Q^k & 0 \\ 0 & (1 - e^{-2\tau \Delta t}) K(\underline{\theta}^b, \underline{\theta}^b) \end{bmatrix}, \quad (2.49)$$

where  $F^k$  and  $Q^k$  are the transition and the process noise matrices of the kinematic state, and  $\tau > 0$  is the forgetting factor that determines the speed of the dynamics for contour estimation [43]. With  $H^k$  being the observation matrix of the kinematic state, the whole observation matrix of the extended state vector is given in cells as:

$$H = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{N_k} \end{bmatrix} \in \mathbb{R}^{2N_k \times (|\underline{x}^k| + N^b)}, \quad (2.50)$$

with

$$H_i = \begin{bmatrix} H^k & K \left( \underline{\theta}_{k,i}, \underline{\theta}^b \right) K(\underline{\theta}^b, \underline{\theta}^b)^{-1} \cdot \cos(\underline{\theta}_{k,i} + \psi_k) \\ & K \left( \underline{\theta}_{k,i}, \underline{\theta}^b \right) K(\underline{\theta}^b, \underline{\theta}^b)^{-1} \cdot \sin(\underline{\theta}_{k,i} + \psi_k) \end{bmatrix} \in \mathbb{R}^{2 \times (|\underline{x}^k| + N^b)} \quad (2.51)$$

corresponds to the observation point at  $\underline{\theta}_{k,i}$  in the local polar coordinate system. With  $R^k$  being the observation noise matrix of the kinematic state, the whole observation noise matrix for the extended state vector is given as:

$$R = \begin{bmatrix} R^k & 0 \\ & \ddots \\ 0 & R^k \end{bmatrix} + \sigma_M^2 I \in \mathbb{R}^{2N_k \times 2N_k}. \quad (2.52)$$



### 3 Forward-Backward Labeled Multi-Bernoulli Smoothing

This chapter presents the proposed methodology of generating reliable tracking results for multiple extended objects, including estimations for both kinematics and contour shapes, from a sequence of LiDAR-recorded point cloud data  $\{Z_k\}_{k=1}^T$ . The term *track* is used in this thesis to denote the sequence of states for the same individual object in the interval between its birth and death, and a multi-object scenario can contain zero, one or several tracks, influenced by object birth/death. The complete tracking process is under the RFS framework and is divided into two sub-processes, namely the forward-backward smoothing and the backward filtering, as illustrated in Fig. 3.1.

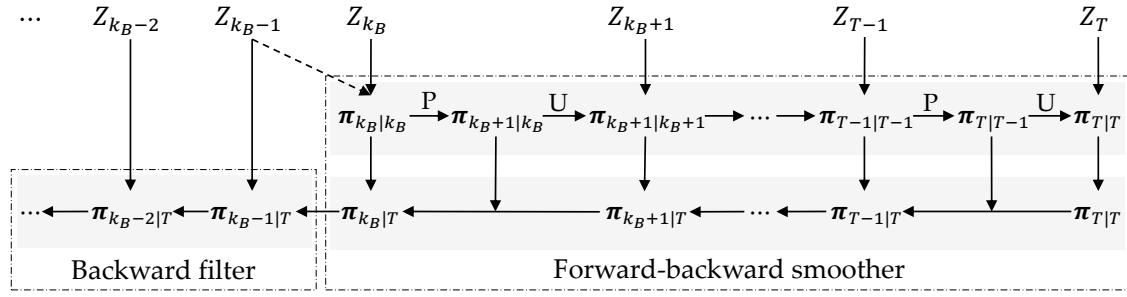


Figure 3.1: The proposed method to generate track-before-detect estimations from a sequence of data  $\{Z_k\}_{k=1}^T$ .  $k_B$  denotes the original birth time of a certain track in the forward pass, and the possibly missing states at  $k_B - 1, k_B - 2, \dots$  will be recovered by the backward filter.

The following sections provide mathematical derivation and implementation details of the process: section 3.1 discusses possible methods that partition the measurement set  $Z_k$  into clusters to reduce the computation consumption for filtering and smoothing; section 3.2 gives details of the forward pass in forward-backward smoothing, while section 3.3 describes the backward pass. The robust birth policy and mixture reduction approach are also included in these two sections. Furthermore, section 3.4 presents the backward filtering process that recovers missing states resulted from the birth policy in the forward pass, thus provides more complete tracks.

### 3.1 Observation Set Partitioning

At each time step of the tracking process, measurement points  $\underline{z}_{k,i} \in Z_k$  should either be regarded as clutter or be associated to exact one object state and participate in the update of this state, i.e., the measurement set  $Z_k$  should be partitioned into exact  $|X| + 1$  clusters including possibly empty clusters. However, the number of possible partitions increases combinatorially as the size of the measurement set grows, thus it is generally not tractable to enumerate all partitions. A more reasonable way is to generate only the most significant partitions to be used in the tracking process while ignoring the partitions that are hardly possible. This section discusses several partitioning methods and justifies the choice of method in this thesis.

#### 3.1.1 K-means

The k-means algorithm is a classic clustering method based on the *Expectation-Maximization* (EM) scheme [64], where "k" in the name denotes the desired number of clusters from the point set. The k-means algorithm minimizes the following loss function:

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \left\| \underline{x}_n - \underline{\mu}_k \right\|^2, \quad (3.1)$$

where  $\underline{\mu}_k$  is the  $k$ -th cluster centroid, and  $r_{nk}$  is the indicator function that associates points to clusters, i.e.,

$$r_{nk} = \begin{cases} 1 & , \text{if } k = \arg \min_j \left\| \underline{x}_n - \underline{\mu}_j \right\|^2, \\ 0 & , \text{else.} \end{cases} \quad (3.2)$$

At the beginning,  $K$  points are chosen according to some rules as the initial cluster centroids, and the process of the k-means algorithm is given by the following EM-steps:

- **Expectation** The indicator function  $r_{nk}$  is updated according to the distances to the new cluster centroids, such that each point is associated to its nearest centroid.
- **Maximization** Keeping  $r_{nk}$  unchanged, it can be immediately concluded by taking the derivatives of (3.1) that the optimal value for  $\underline{\mu}_k$  is

$$\underline{\mu}_k = \frac{\sum_n r_{nk} \underline{x}_n}{\sum_n r_{nk}}, \quad (3.3)$$

i.e., the mean position of all points associated to this cluster, thus the name "means".

These steps are executed recursively until convergence or meet of other termination conditions. According to the nature of EM, the clustering results of k-means can converge to saddles points or local minimum where the derivative of (3.1) is zero. Thus, the choice of initial centroids is crucial to obtain better clusters. While they can be randomly chosen, a wiser way is to select the initial centroids to be possibly sparse, i.e., select the most distant  $k$  points. This strategy is proposed as *k-means++*, and can yield considerable improvement in the final error of k-means [65].

### 3.1.2 Distance Partitioning

Taking into account the characteristics of the points' spatial distribution in the measurement set, another clustering method is proposed in [66], known as distance partitioning. As the name implies, the measurement set is partitioned according to the point-to-point distances based on the fact that measurements originated from the same extended object are spatially close to each other. The process of distance partitioning is straightforward: given the measurement set  $Z$ , the pairwise distances  $\{d_{ij}\}_{i \neq j}$  among the point are first calculated, from which a set of distance thresholds  $\{D_i\}_{i=1}^{N_D}$  with  $D_i < D_{i+1}$  is selected according to some rule. For each  $D_i$ , the partitions are computed such that every pair of points with distance smaller than  $D_i$  will belong to the same cluster. Thus, we get  $N_D$  partitions, and since the  $D_i$  are in the increasing order, each partition contains non-increasing number of clusters, and the clusters typically contain more measurements.

One problem with distance partitioning method is that measurements from two or more spatially close objects might be assigned to the same cluster if the distance threshold is too large. Neither can it exclude clutter points that are close to the measurements. Originally in [66], the Mahalanobis distance based on noise covariance is suggested as the distance metric, and a  $\chi^2$  distribution for the distances is used to set the lower and upper bounds of the distance thresholds. An extra sub-partition process is proposed in [67] to further divide a cluster into smaller clusters when it contains more points than expected, which solves the problem caused by closely spaced objects.

### 3.1.3 Prediction Partitioning

By incorporating the predicted object state from the last time step, the prediction partitioning method is proposed in [35], where the likelihood of measurements based on the prediction from the previous time step is utilized to divide the current measurement set. A point is associated to one object if the corresponding likelihood is larger than the threshold, and will be regarded as clutter if not associated to any object. This method works well when the true object motion can be well modeled by the dynamic motion model, and is expected to be insufficient when the objects take rapid maneuvers.

### 3.1.4 Comparison of Partitioning Methods

The scenario in Fig. 3.2a demonstrates the characteristics of the above discussed partitioning methods, where the cluttered measurements and the two corresponding predicted contours are presented. Since the measurements come from LiDAR-recorded maneuvering objects, the prediction partitioning method is only appended additionally to the distance partitioning method to prune the results rather than tested as an individual method. The results are shown in Fig. 3.2.

The incorrect result in Fig. 3.2b on this scenario is typical for k-means++, because for this type of scenario where a large number of measurements are densely distributed in

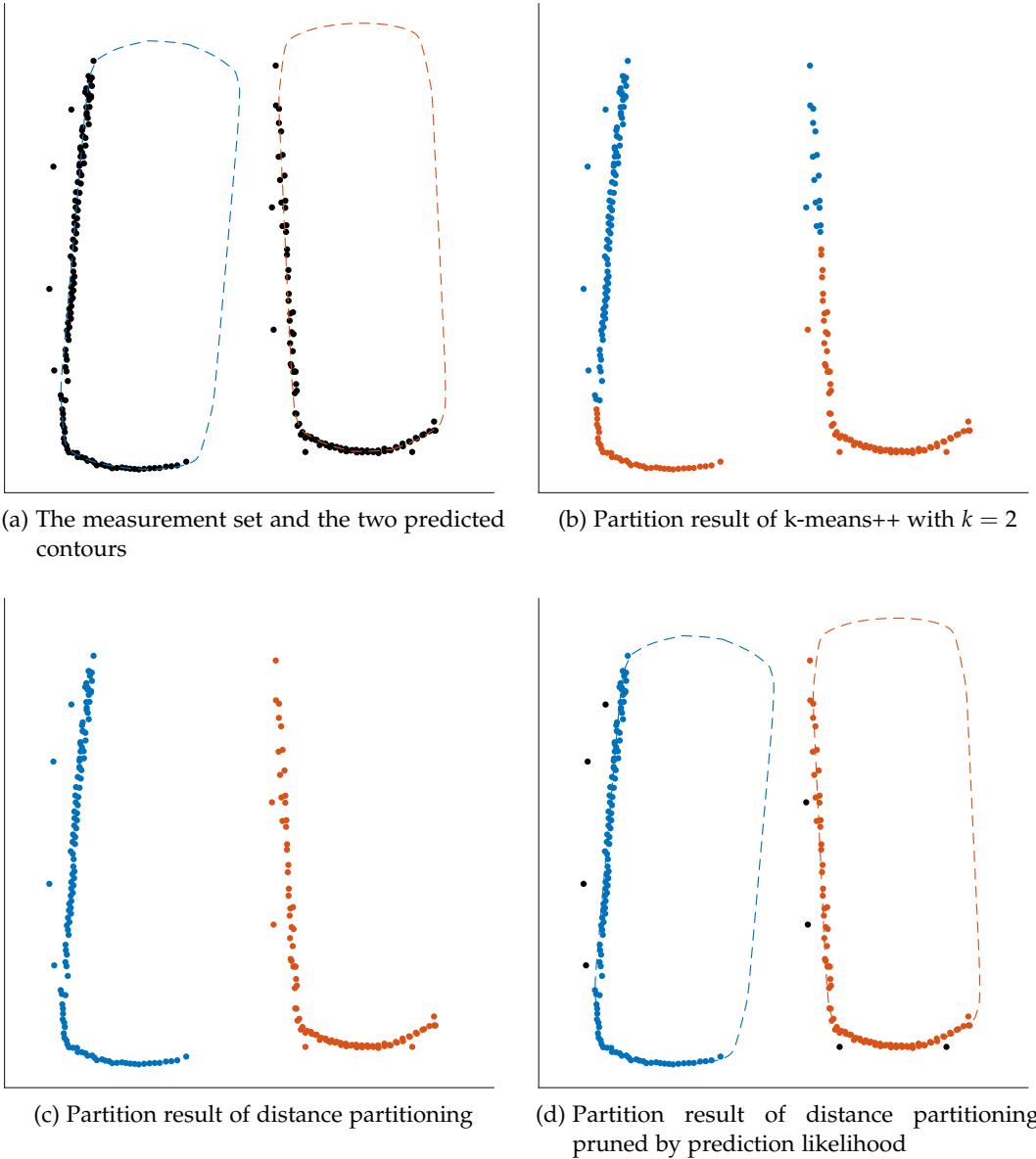


Figure 3.2: Comparison of partition results from different clustering methods. As can be observed from the figures, k-means++ will mis-partition the measurement set when the two objects are close to each other, distance partitioning provides correct partitions but cannot exclude noise points that are close to the contour, which can be fixed by prediction pruning.

one part of the surveillance area (in this case the bottom part), the k-means loss function profits much more by dividing these measurements into the same cluster rather than splitting them. This problem is then solved by the distance partitioning method as

shown in Fig. 3.2c, whereas the clutter points are also partitioned into both clusters. Further pruning the result according to the likelihood based on the predicted contours finally excludes the clutter points from the clusters and yields desired partitions (Fig. 3.2d). The reason that prediction partitioning can be used here as pruning condition for maneuvering objects is that the frame rate of LiDAR is relatively high compared to the time needed for the influence of maneuvering to become obvious. Because of the superior performance of distance partitioning with prediction likelihood pruning, it is also used in this thesis in the following sections.

## 3.2 Forward Filtering

This section introduces the forward pass (filtering) of multiple extended object tracking based on the LMB filter and the Gaussian process for star-convex shape model. The multi-object state is model as an LMB RFS (see section 2.3.2) on the state space  $\mathbb{X}$  and label space  $\mathbb{L}$ , given by the parameter set

$$\pi = \left\{ \left( r^{(\ell)}, p^{(\ell)} \right) \right\}_{\ell \in \mathbb{L}} \quad (3.4)$$

where each unique label  $\ell$  identifies one individual track. The probability density function of the LMB RFS is propagated over time with the Bayesian multi-object inference process (section 2.3.3). The forward computation consists of two steps: the prediction step incorporates newborn objects and surviving objects from the last time step; the update step adjusts the predictions with the current measurement set. Besides, the estimated states are extracted by some estimator from the posterior density, and the proposed robust birth policy is also discussed in this section.

### 3.2.1 Prediction Step

Suppose that the posterior multi-object density from the last time step and the multi-object birth model of the current time step are both LMB RFSs defined on  $\mathbb{X} \times \mathbb{L}$  and  $\mathbb{X} \times \mathbb{B}$ , respectively, and are given by the following parameter sets

$$\pi = \left\{ \left( r^{(\ell)}, p^{(\ell)} \right) \right\}_{\ell \in \mathbb{L}}, \quad \pi_B = \left\{ \left( r_B^{(\ell)}, p_B^{(\ell)} \right) \right\}_{\ell \in \mathbb{B}} \quad (3.5)$$

Consequently, the corresponding probability density functions are:

$$\begin{aligned} \pi(\mathbf{X}) &= \Delta(\mathbf{X}) w(\mathcal{L}(\mathbf{X})) p^{\mathbf{X}} \\ \pi_B(\mathbf{X}) &= \Delta(\mathbf{X}) w_B(\mathcal{L}(\mathbf{X})) [p_B]^{\mathbf{X}} \end{aligned} \quad (3.6)$$

with

$$\begin{aligned} w(L) &= \prod_{i \in \mathbb{L}} \left(1 - r^{(i)}\right) \prod_{\ell \in L} \frac{1_{\mathbb{L}}(\ell) r^{(\ell)}}{1 - r^{(\ell)}} \\ w_B(L) &= \prod_{i \in \mathbb{B}} \left(1 - r_B^{(i)}\right) \prod_{\ell \in L} \frac{1_{\mathbb{B}}(\ell) r_B^{(\ell)}}{1 - r_B^{(\ell)}} \\ p(x, \ell) &= p^{(\ell)}(x) \\ p_B(x, \ell) &= p_B^{(\ell)}(x) \end{aligned} \quad (3.7)$$

Recall that  $\Delta(\mathbf{X})$  is the indicator function for distinct labels, and  $\mathcal{L}(\mathbf{X})$  is the distinct label set of  $\mathbf{X}$ . Since an LMB is a special case of a GLMB (with one single term), it follows from proposition 8 in [19] that the predicted multi-object density incorporating  $\pi$  and  $\pi_B$  is also in the form of a GLMB. A stronger result is given in [21], stating that the predicted multi-object density is also an LMB RFS on the state space  $\mathbb{X}$  and the label space  $\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}$  defined by the parameter set:

$$\pi_+ = \left\{ \left(r_{+,s}^{(\ell)}, p_{+,s}^{(\ell)}\right) \right\}_{\ell \in \mathbb{L}} \cup \left\{ \left(r_B^{(\ell)}, p_B^{(\ell)}\right) \right\}_{\ell \in \mathbb{B}} \quad (3.8)$$

where

$$\begin{aligned} r_{+,s}^{(\ell)} &= \eta_s(\ell) r^{(\ell)}, \\ p_{+,s}(\underline{x}, \ell) &= \frac{\langle p_s(\cdot, \ell) f(\underline{x}|\cdot, \ell), p(\cdot, \ell) \rangle}{\eta_s(\ell)}, \\ \eta_s(\ell) &= \langle p_s(\cdot, \ell), p(\cdot, \ell) \rangle. \end{aligned} \quad (3.9)$$

$p_s(\underline{x}, \ell)$  is the state-dependent single object survival probability that controls the survival/death behavior of an object to the next time step,  $\eta_s(\ell)$  the survival probability of a certain track  $\ell$ , and  $f(\underline{x}|\cdot, \ell)$  the single track transition density of track  $\ell$ . For the surviving tracks from the last time step, the predicted label is the same as the previous, and the predicted existence probability and spatial distribution are re-weighted by the survival probability and transition density. For the newborn tracks, the labels are new distinct labels from  $\mathbb{B}$  satisfying  $\mathbb{B} \cap \mathbb{L} = \emptyset$ . Thus, the prediction step incorporates the posterior multi-object density from the last time step and the multi-object birth density for newborn objects while still preserving the form of an LMB RFS.

### 3.2.2 Robust Birth Policy

In the prediction step, a dynamic birth model  $\pi_B$  is realized by a consecutive process among neighboring time steps resembling the adaptive birth density proposed in [21]. The idea lies in that the measurements from the last time steps not assigned to any track are more likely to correspond to new objects. As for point objects, this strategy can be easily applied by merely considering all unassociated measurement points from the last steps as birth candidates and initialize for each of them a mean state with zero velocity and the corresponding existence probability. This, however, is expected to be insufficient

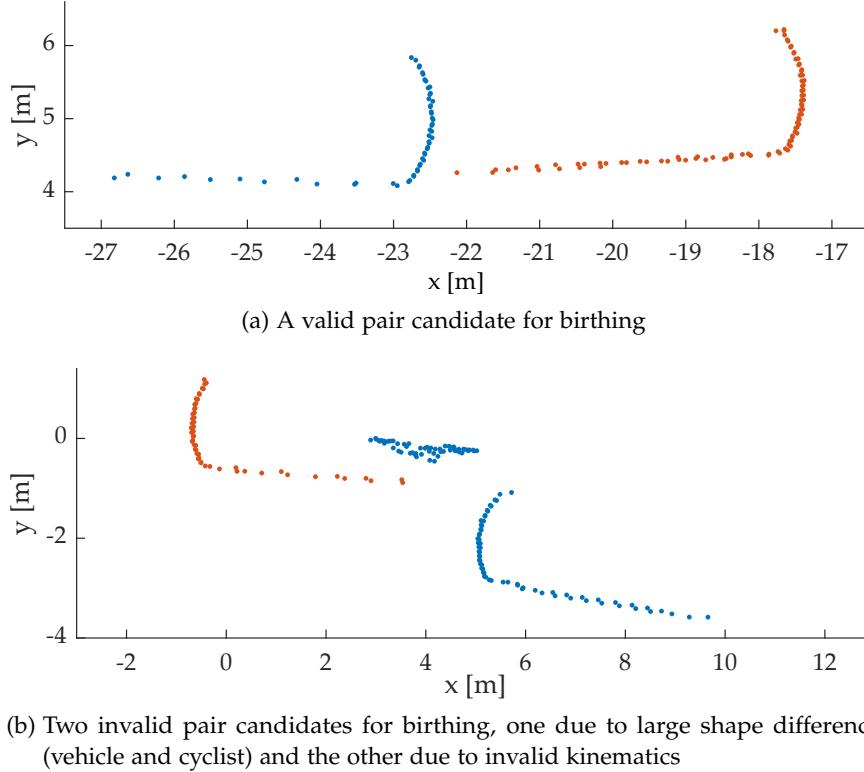


Figure 3.3: Examples of pair candidates for birthing. Different colors denote clusters from different time steps.

for extended objects, since the choice of reference point, the kinematics and the contour shape require more accurate initialization.

Due to the high frame rate of LiDAR (usually near 10Hz for common commercial LiDARs), the cluster shape formed by the measurements from the same object can be considered as nearly unchanged between two adjacent steps. Thus, to enhance the accuracy of state initialization, cluster pairs are enumerated among unassigned clusters from two different previous time steps as candidates instead of merely using individual clusters. The Iterative Closest Point algorithm [68] with the Huber robust loss function [69] is employed to register between the two clusters of each pair candidate, and new objects are initialized by the valid pairs yielding high shape similarity and valid kinematics, as illustrated in Fig. 3.3. Note that in Fig. 3.3b, the kinematics from the two vehicle-like clusters are invalid since the object motion is assumed to be constrained only in the direction of the orientation. All unassociated clusters are kept for  $M$  time steps and discarded afterwards, so that the enumeration of pair candidates requires less computation and the initial estimation of kinematics and contour shape suffers less from the accumulated error over time.

Typically, there are two possible types of contour shape from the point cloud recorded by a one-layer LiDAR, namely the I-shape and the L-shape, as illustrated in Fig. 3.4. The

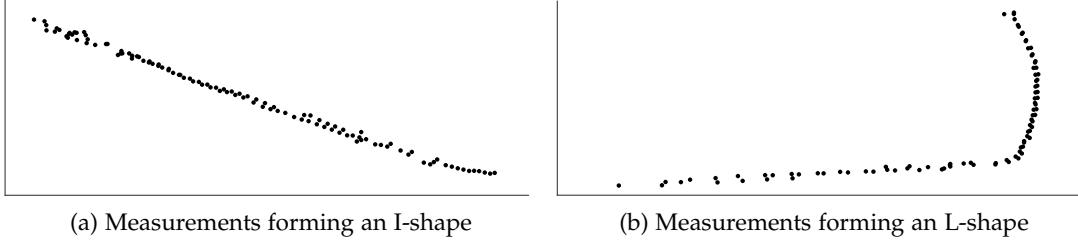


Figure 3.4: Two types of possible shapes formed by measurement points from LiDAR observation.

I-shape corresponds to objects that are almost vertical to the connecting line between the object center and the sensor and has only one side visible from the view of the sensor; while the L-shape corresponds to objects that have two visible sides. Small amount of measurements from other sides may also appear over the corner, but can be categorized into these two types and similarly processed. In general, an L-shaped cluster can provide more information on the object contour than an I-shaped cluster, since it contains points from two sides of the object and thus information on two geometric dimensions. Besides, the number of points in a cluster can also influence its quality as a candidate for birthing.

Unlike tracking point objects, where a single measurement contains all information required for object initialization, tracking extended objects requires a cluster of measurements containing enough information to yield robust initial contour estimation beside kinematics. This requirement is typically not fulfilled in the sensor data for objects approaching from distant positions, since the density of measurement points usually decreases as the distance to the sensor grows larger. Consequently, unreliable initialization will lead to inaccurate initialization and negative influence on the subsequent inference processes.

As mentioned above, the quality of a cluster as a candidate for birthing is mainly decided by the number and the spatial distribution of measurement points in it. Thus, in this thesis, a measure is introduced for the suitability of a cluster regarding birthing by three factors: its cardinality, its mean distance  $s$  to the sensor, and its shape ratio calculated by  $V = \sigma_2/\sigma_1$ , where  $\sigma_1^2$  and  $\sigma_2^2$  are the two variances from the Principle Component Analysis (PCA) of cluster points with  $\sigma_1^2 \geq \sigma_2^2$ , as illustrated in Fig. 3.5. A *robust birth policy* then adjusts the existence probability of the new LMB track initialized from the cluster based on the horizontal angular resolution  $D$  (unit in deg) of the sensor and a constant parameter  $\alpha > 0$ , i.e.,

$$r'_B^{(\ell)} = r_B^{(\ell)} \cdot \left( t \cdot \frac{D \cdot |\mathcal{U}_{\theta(\ell)}(Z)|}{\varphi(\mathcal{U}_{\theta(\ell)}(Z))} + (1 - t) \cdot V(\mathcal{U}_{\theta(\ell)}(Z)) \right), \quad (3.10)$$

where  $\mathcal{U}(Z)$  denotes a certain partition of the measurement set  $Z$  into clusters,  $\theta : \mathbb{L} \rightarrow \{0, 1, \dots, |\mathcal{U}(Z)|\}$  denotes the association map that assigns the labels on the label space

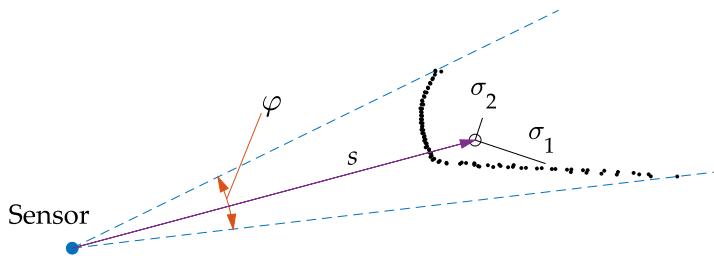


Figure 3.5: Factors influencing the suitability of a cluster regarding birthing: measurement density (calculated by the cluster cardinality and the azimuth angle range  $\varphi$ ), mean distance  $s$  to the sensor, and shape ratio (represented by the PCA variances  $\sigma_1^2, \sigma_2^2$ ).

$\mathbb{L}$  (thus the tracks) to clusters with  $\theta(\ell) = 0$  denoting the misdetection of track  $\ell$ , and  $\varphi(\mathcal{U}_{\theta(\ell)}(Z))$  the azimuth angle range of the cluster  $\mathcal{U}_{\theta(\ell)}(Z)$ ) from the view of the sensor. The factor  $t = e^{-\alpha s(\mathcal{U}_{\theta(\ell)}(Z))}$  controls the weights of the two terms based on the cluster's mean distance to the sensor, of which the first term rewards clusters with a high measurement density and the second term rewards clusters with measurements from different dimensions of an object. Although possibly dense clusters with measurements over more dimensions of the object contour are preferred, the observation of a single side can still be sufficient if the object is close to the sensor (the ego-vehicle) and the span of this dimension can be relatively accurately reflected by the measurements, e.g., to account for car-following maneuvers or parallel motion. Fig. 3.6 shows two examples of clusters that receive high existence probability from the quality measure.

Higher quality of initialization can be achieved in such manner at the cost of losing some early states due to insignificant existence probability of certain candidates, which will be discussed later in the proposed backward filtering process.

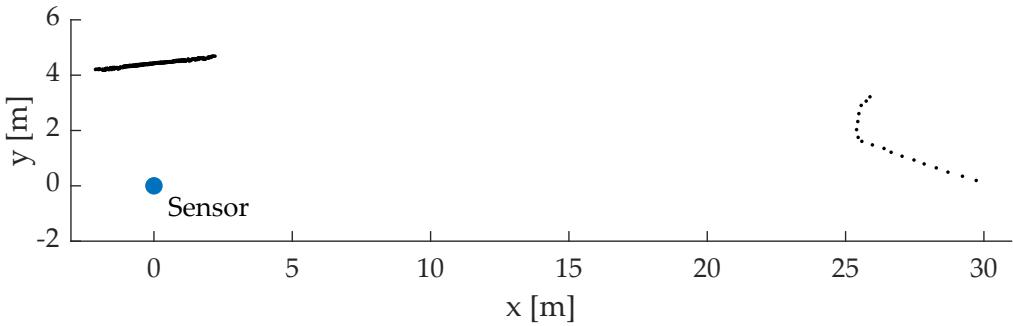


Figure 3.6: Examples of clusters receiving high existence probability from the quality measure, either due to high measurement density (the left one) or large shape ratio (the right one).

### 3.2.3 Multiple Extended Object Likelihood

The LMB filter for extended objects differs from the classic single-object one mainly in the partition-based multiple extended object likelihood in the update step instead of the point-based, since various partitions and association maps should also be taken into account. Let  $p_D(\underline{x}, \ell)$  be the state-dependent detection probability for an individual track  $\ell$ ,  $\kappa$  be the detection-independent clutter density, and  $g_C(Z)$  be the clutter distribution, the multiple extended object likelihood of measurement set  $Z$  based on the LMB RFS  $X$  is given by [36]

$$g(Z|X) = g_C(Z) \sum_{i=1}^{|X|+1} \sum_{\substack{\mathcal{U}(Z) \in \mathcal{P}_i(Z) \\ \theta \in \Theta(\mathcal{U}(Z))}} \left[ \psi_{\mathcal{U}(Z)}(\cdot; \theta) \right]^X, \quad (3.11)$$

where

$$\psi_{\mathcal{U}(Z)}(\underline{x}, \ell; \theta) = \begin{cases} \frac{p_D(\underline{x}, \ell) \tilde{g}(\mathcal{U}_{\theta(\ell)}(Z)|\underline{x}, \ell)}{[\kappa]^{\mathcal{U}_{\theta(\ell)}(Z)}}, & \theta(\ell) > 0, \\ 1 - p_D(\underline{x}, \ell), & \theta(\ell) = 0, \end{cases} \quad (3.12)$$

is the likelihood based on a single Bernoulli track  $\ell$  and the association map  $\theta$ ,  $\mathcal{P}_i$  is the set of partitions dividing the measurement set  $Z$  into exact  $i$  clusters,  $\Theta(\mathcal{U})$  the space of possible track-to-cluster association maps under partition  $\mathcal{U}$ ,  $\tilde{g}(\mathcal{U}_{\theta(\ell)}(Z)|\underline{x}, \ell) = [g(\cdot|\underline{x}, \ell)]^{\mathcal{U}_{\theta(\ell)}(Z)}$  the likelihood for cluster  $\mathcal{U}_{\theta(\ell)}(Z)$  based on single state  $(\underline{x}, \ell)$  that is independent of other objects. In this sense, the multiple extended object likelihood incorporates misdetection and clutter or false alarms in the measurements, and also models the data association uncertainty between the measurement set and the multi-object state.

### 3.2.4 Update Step

As mentioned in section 2.3.2, an LMB RFS comprises statistically independent tracks. However, since the existence of one track can affect the association probabilities for other tracks in the update process, the measurement updated tracks are no longer statistically independent. Thus, an LMB RFS can not exactly represent the multi-object posterior after the measurement update. In contrast, the GLMB RFS facilitates multiple realizations (hypotheses) for a set of track labels  $L$  and can naturally represent the data association uncertainties in update by generating a hypothesis  $\xi$  for each possible track-to-cluster association. Hence, a GLMB allows for the exact representation of the statistically dependent tracks after the measurement update [19]. To possibly reduce the approximation error in the update process while still preserving the form of LMB for its computational efficiency, the predicted LMB from (3.8) is interpreted as a GLMB to enable exact measurement updates, which is then approximated by an LMB after the measurement update.

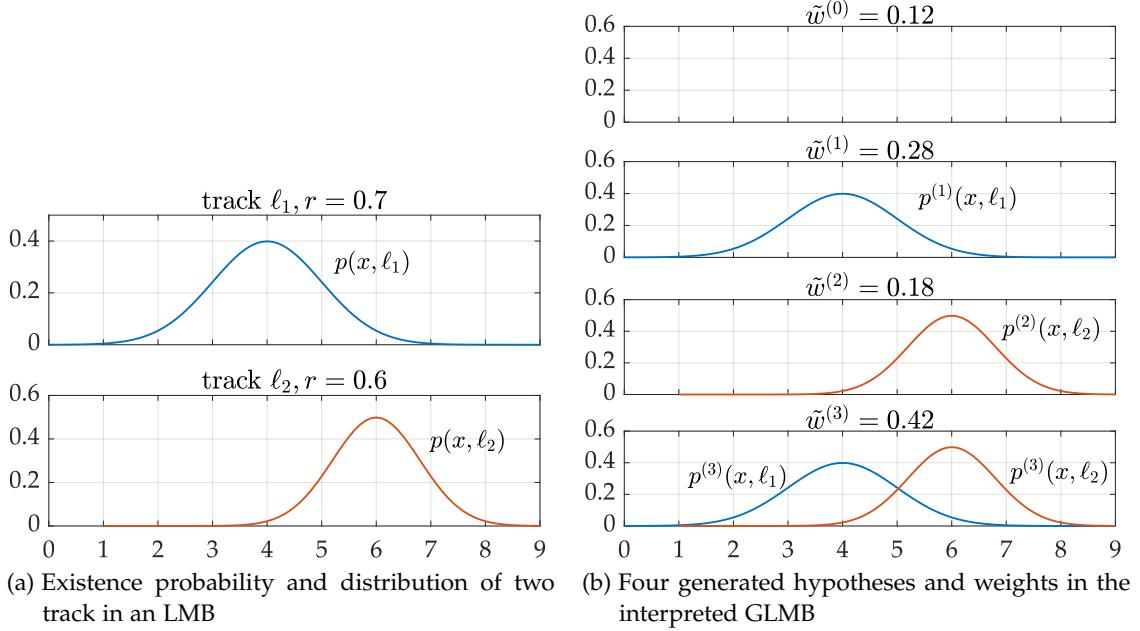


Figure 3.7: Example of hypothesis generation in the LMB-to-GLMB interpretation. The probability distributions of each hypothesis inherit from the involved tracks, and the corresponding hypothesis weight is calculated by multiplying the existence probabilities of all involved tracks.

The interpretation from LMB to GLMB is exact and straightforward: hypotheses can be simply generated by enumerating the presence of each track (label), and the corresponding weights can be calculated by multiplying all involved existence probabilities:

$$\tilde{w}_+^{(\xi)} = \prod_{\ell \in \mathbb{L}} (1 - r_+^{(\ell)}) \prod_{\ell \in \mathcal{L}(\xi)} \frac{r_+^{(\ell)}}{1 - r_+^{(\ell)}}, \quad (3.13)$$

as shown in the example in Fig. 3.7. This process, however, can give rise to huge amount of computation demand since the number of hypotheses grows exponentially as tracks increase (more specifically, an LMB with  $N$  tracks can be interpreted into a GLMB with  $2^N$  hypotheses), which is in general intractable. A more viable approximation can be obtained by only considering the hypotheses with the most significant weights, which requires weight-based ordered hypothesis generation without exhaustive enumeration. A solution is proposed in [20] for this task to construct a directed weighted graph and apply the k-shortest path algorithm to find the  $k$  most significant hypotheses, as illustrated in Fig. 3.8 with  $I = \mathcal{L}(\xi)$  and the node costs calculated as:

$$C^{(I, \xi)}(\ell_j) = -\ln \frac{r_+^{(\ell_j)}}{1 - r_+^{(\ell_j)}}. \quad (3.14)$$

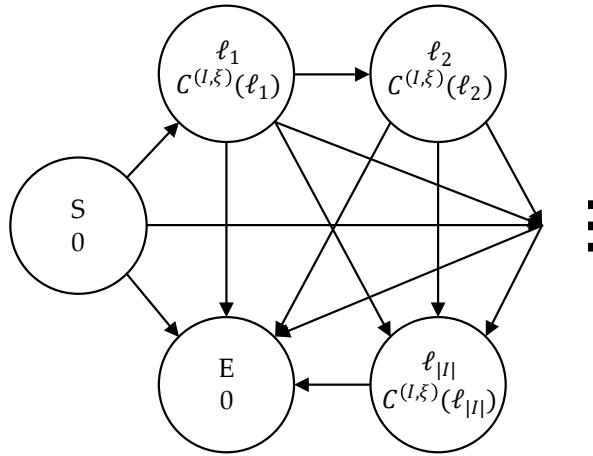


Figure 3.8: Construction of a k-shortest path problem to generate the most significant hypotheses without exhaustive enumeration [20]. The presence of the node  $\ell$  on the path from  $S$  to  $E$  indicates its presence in the corresponding hypothesis.

After the LMB-to-GLMB interpretation, an exact GLMB update can be performed within each hypothesis by correcting the states with possible partitions and association maps and adjusting the hypothesis weight by the multi-object likelihood, i.e., [19]

$$\begin{aligned} p^{(\theta)}(\underline{x}, \ell | Z) &= \frac{p_+(\underline{x}, \ell) \psi_Z(\underline{x}, \ell; \theta)}{\eta_Z^{(\theta)}(\ell)}, \\ w^{(\xi, \theta)}(Z) &\propto \tilde{w}_+^{(\xi)}(\mathcal{L}(\xi)) \left[ \eta_Z^{(\theta)} \right]^{\mathcal{L}(\xi)}, \\ \eta_Z^{(\theta)}(\ell) &= \langle p_+(\cdot, \ell), \psi_Z(\cdot, \ell; \theta) \rangle, \end{aligned} \quad (3.15)$$

where  $\theta$  is a certain track-to-cluster association map,  $\xi$  is the hypothesis index, and  $p_+(\underline{x}, \ell) = p_+^{(\ell)}(\underline{x})$  is the predicted state probability distribution of track  $\ell$  as in (3.8).

A ranked assignment algorithm is also utilized to produce only the  $N_A$  most substantial track-to-cluster association maps within each hypothesis based on the current partitions [36]. Consequently, the number of new hypotheses after update is given by  $N_H = |\Xi| \cdot \left| \bigcup_{i=1}^{|X|+1} \mathcal{P}_i \right| \cdot N_A$  and will rapidly explode. This problem can be avoided by approximating the posterior GLMB again as an LMB, i.e., by assuming that the tracks are once again independent of each other. The approximation can be executed by summing up the weights and the weighted probability distributions among all hypotheses where a certain track is present, as exemplified in Fig. 3.9. Normally, if the prior probability distribution of each track is assumed to be Gaussian before prediction, then the corresponding posterior distribution will be in the form of a Gaussian mixture with  $N_H/2$  components. Again, the Gaussian components with minor weights can be pruned for each track to save storage and computation.

Furthermore, in many real-world applications of extended object tracking, hypotheses containing wrong track-to-cluster associations (if any, after prediction pruning discussed

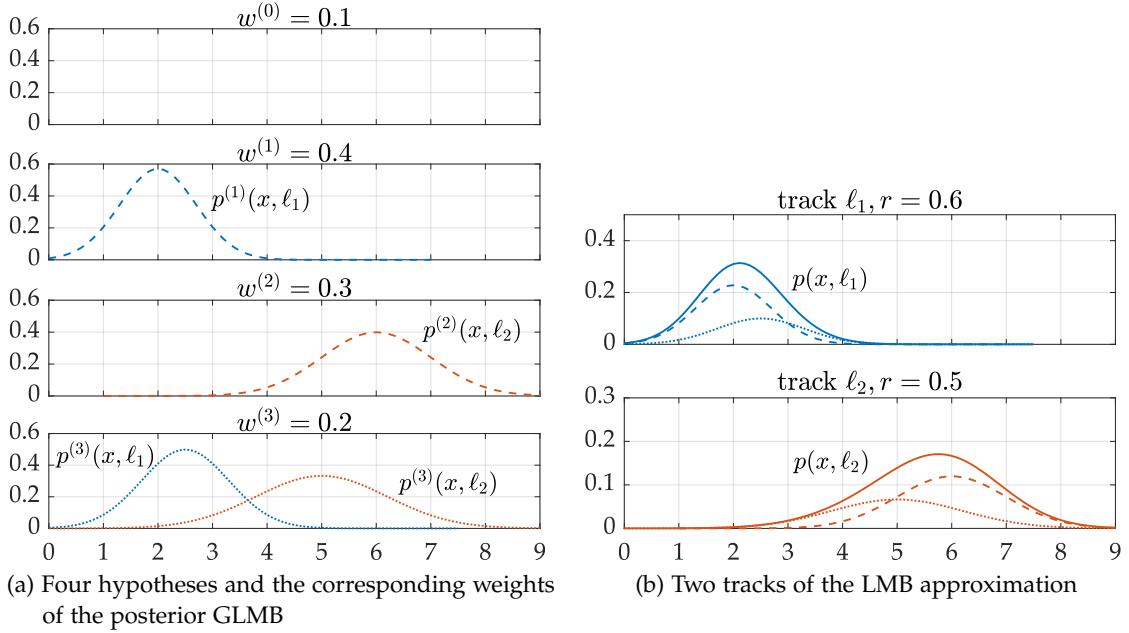


Figure 3.9: Example of the GLMB-to-LMB approximation. The probability distribution of each track in the LMB approximation is just the weighted sum of the distributions of the involved hypotheses in the GLMB, and the existence probability is calculated by summing up the corresponding weights.

in section 3.1) will hardly survive due to insignificant likelihood that quickly approaches zero as the distance deviation from the predicted position increases. This can be explained by the high frame rate of LiDAR and the smooth radial function of the contour estimation from the recursive Gaussian process, i.e., the prediction error caused by maneuvers has not yet become too large, and an object contour comprises no "sharp corner". The mixture reduction algorithm based on the Kullback-Leibler divergence [70] can be used to guarantee that only the most significant component remains, thus not only reduces the computational demand in the subsequent steps, but also enables simple track smoothing that will be discussed later. Alternatively, a particle filter can be applied instead of propagating the Gaussian mixtures, as investigated in [20].

Finally, after the LMB-to-GLMB interpretation for precise measurement update and the GLMB-to-LMB approximation for the reduction of computational cost, the posterior LMB can be calculated based on the predicted LMB parameter set (3.8) as [36]:

$$r^{(\ell)} = \sum_{L \subseteq \mathbb{L}_+} \sum_{i=1}^{|L|+1} \sum_{\substack{\mathcal{U}(Z) \in \mathcal{P}_i(Z) \\ \theta \in \Theta(\mathcal{U}(Z))}} 1_L(\ell) w_{\mathcal{U}(Z)}^{(\theta)}(L), \quad (3.16)$$

$$p^{(\ell)}(\underline{x}) = \frac{1}{r^{(\ell)}} \sum_{L \subseteq \mathbb{L}_+} \sum_{i=1}^{|L|+1} \sum_{\substack{\mathcal{U}(Z) \in \mathcal{P}_i(Z) \\ \theta \in \Theta(\mathcal{U}(Z))}} 1_L(\ell) w_{\mathcal{U}(Z)}^{(\theta)}(L) p^{(\theta)}(\underline{x}, \ell),$$

where

$$\begin{aligned} w_{\mathcal{U}(Z)}^{(\theta)}(L) &\propto w_+(L) \left[ \eta_{\mathcal{U}(Z)}^{(\theta)} \right]^L, \\ p^{(\theta)}(\underline{x}, \ell | \mathcal{U}(Z)) &= \frac{p_+(\underline{x}, \ell) \psi_{\mathcal{U}(Z)}(\underline{x}, \ell; \theta)}{\eta_{\mathcal{U}(Z)}^{(\theta)}(\ell)}, \\ \eta_{\mathcal{U}(Z)}^{(\theta)}(\ell) &= \left\langle p_+(\cdot, \ell), \psi_{\mathcal{U}(Z)}(\cdot, \ell; \theta) \right\rangle. \end{aligned} \quad (3.17)$$

Note the structure similarity between (2.28) and (3.13) as well as between (3.15) and (3.17). The first pair shows that the interpretation from the predicted LMB to GLMB is to simply construct a GLMB with only one term in the hypothesis index set  $\Xi$ ; the second pair shows that the GLMB-to-LMB approximation is calculated by just combining the weights and probability distributions of the involved hypotheses for each track under the assumption that all tracks are independent.

### 3.2.5 State Extraction

After the *Predict-Update* processes of each time step, the current cardinality and the individual estimated object states can be extracted from the posterior multi-object density. While theoretical Bayes-optimal solutions to the state estimator exist [11], their computation is difficult to realize. Instead, two simple and intuitive sub-optimal estimators can be applied for this objective, as will be discussed below.

Since the posterior density is still in the form of an LMB RFS, the simplest way of state extraction is to check all updated tracks by their existence probabilities using an application-specified threshold  $r_E$ , and extract the estimated state  $(\hat{\underline{x}}, \ell)$  of each track satisfying  $r^{(\ell)} > r_E$ , i.e., the estimated set of object states is given by

$$\hat{\mathbf{X}} = \left\{ (\hat{\underline{x}}, \ell) : r^{(\ell)} > r_E \right\}_{\ell \in \mathbb{L}}, \quad (3.18)$$

where  $\hat{\underline{x}}$  is typically the mean of a multivariate Gaussian or of the Gaussian component with highest weight in an Gaussian mixture. A lower threshold for existence probability will lead to immediate report of a new track yet higher number of clutter tracks, while a higher threshold will result in fewer clutter tracks yet also more conservative extraction of tracks. With detection probability  $p_D \approx 1$ , the existence probability of a previously confirmed track can be considerably reduced by a single misdetection, which can bring about cardinality error when the threshold is high. Instead, a hysteresis can be used rather than a single threshold to allow extraction of previously confirmed track with lower existence probability [21], i.e.,

$$\hat{\mathbf{X}} = \left\{ (\hat{\underline{x}}, \ell) : r^{(\ell)} > r_{E,l} \wedge r_{\max}^{(\ell)} > r_{E,u} \right\}_{\ell \in \mathbb{L}}, \quad (3.19)$$

where  $r_{\max}^{(\ell)}$  denotes the highest "history" existence probability of track  $\ell$ , and  $r_{E,l} < r_{E,u}$  are the lower and upper hysteresis thresholds. This guarantees that if the existence probability of a track has once reached the upper threshold, it can be accepted with a relatively lower threshold afterwards.

An alternative estimator utilizes not only the posterior LMB density, but also the posterior GLMB density before approximation to first calculate the maximum *a posteriori* (MAP) cardinality and then extract the corresponding number of tracks with highest existence probability [20]. Let  $C$  denote the new index set for the posterior GLMB in (3.15) where for each pair  $(\xi, \theta)$  there is a unique corresponding index  $c \in C$ , the existence probability of a track  $\ell$  then follows (3.16) as:

$$r^{(\ell)} = \sum_{c \in C} 1_{\mathcal{L}(c)}(\ell) w^{(c)}, \quad (3.20)$$

and the cardinality distribution is given as:

$$\rho(n) = \sum_{c \in C} \delta_{|\mathcal{L}(c)|}(n) w^{(c)}, \quad (3.21)$$

where  $\delta_x(y)$  is the Dirac delta function that returns 1 if  $x = y$  and 0 otherwise. The MAP cardinality can then be obtained as  $\hat{N} = \arg \max \rho$ , and the  $\hat{N}$  tracks in the posterior LMB density with the highest existence probabilities are extracted as the multi-object state estimation at the current time step.

In this thesis, the MAP state extractor is implemented and integrated, since it requires no manual specification of thresholds for existence probability but rather makes better use of the information in the posterior distribution, thus suitable for scenarios containing objects of various geometries.

### 3.3 Backward Smoothing

The complete processes of the forward pass described above follows an online manner, i.e., for the calculation of the posterior at each time step, only the previous and the current measurement sets are required. Thus, the forward pass can be applied in real-time as the measurements are coming in batches (e.g., a sequence of LiDAR-recorded point cloud frames). The results of the forward pass can be further improved in a backward pass by post-processing all the extracted states incorporating the future data up to the last time step  $T$ , namely the smoothing process as in (2.33). Since the complete dataset is demanded for this operation including from the future, the backward pass can only be executed offline.

The smoothing process is carried out for each individual track from time  $T_D^{(\ell)}$  and runs backward until time step  $k_B^{(\ell)}$ , where  $T_D^{(\ell)}$  is the confirmed death time of track  $\ell$  (i.e., pruned due to too low existence probability) and  $k_B^{(\ell)}$  the birth time in the forward pass. Since the Bernoulli RFS is closed under the smoothing operation [28], the equations for the backward pass can be easily obtained by inserting the posterior LMB density (3.16) into the multi-object Bayes smoothing equation (2.36). Moreover, let  $\pi_{k|T} = \{(r_{k|T}^{(\ell)}, p_{k|T}^{(\ell)})\}_{\ell \in \mathbb{L}}$  denote the smoothed LMB density from time  $T$  to  $k$  and assume no re-entry of track after death, the smoothed density from time  $T$  to  $k - 1$  is

also an LMB RFS  $\pi_{k-1|T} = \left\{ (r_{k-1|T}^{(\ell)}, p_{k-1|T}^{(\ell)}) \right\}_{\ell \in \mathbb{L}}$  given by the parameters:

$$\begin{aligned} r_{k-1|T}^{(\ell)} &= 1 - \frac{\left(1 - r_{k-1|k-1}^{(\ell)}\right) \left(1 - r_{k|T}^{(\ell)}\right)}{\left(1 - r_{k|k-1}^{(\ell)}\right)}, \\ p_{k-1|T}^{(\ell)}(\underline{x}) &= p_{k-1|k-1}(\underline{x}, \ell) \times \left( \alpha_{S,k|T}(\underline{x}, \ell) \right. \\ &\quad \left. + \beta_{S,k|T}(\underline{x}, \ell) \left\langle f_{k|k-1}(\cdot | \underline{x}, \ell), \frac{p_{k|T}(\cdot, \ell)}{p_{k|k-1}(\cdot, \ell)} \right\rangle \right) / \gamma_{S,k|T}(\ell), \end{aligned} \quad (3.22)$$

where

$$\begin{aligned} \alpha_{S,k|T}(\underline{x}) &= (1 - p_{S,k|k-1}(\underline{x}, \ell)) \frac{(1 - r_{k|T})}{(1 - r_{k|k-1})}, \\ \beta_{S,k|T}(\underline{x}) &= p_{S,k|k-1}(\underline{x}, \ell) \frac{r_{k|T}}{r_{k|k-1}}, \\ \gamma_{S,k|T}(\ell) &= \left\langle p_{k-1|k-1}(\cdot, \ell), \alpha_{S,k|T}(\cdot, \ell) \right. \\ &\quad \left. + \beta_{S,k|T}(\cdot, \ell) \left\langle f_{k|k-1}(\cdot | \cdot, \ell), \frac{p_{k|T}(\cdot, \ell)}{p_{k|k-1}(\cdot, \ell)} \right\rangle \right\rangle, \end{aligned} \quad (3.23)$$

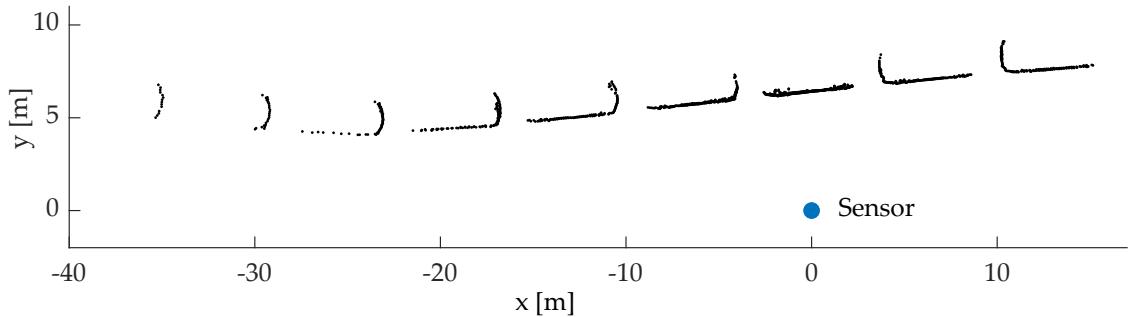
and for each track  $\ell$ , the recursion is initialized by the parameter set

$$\left( r_{T_D^{(\ell)}|T}^{(\ell)}, p_{T_D^{(\ell)}|T}^{(\ell)} \right) = \left( r_{T_D^{(\ell)}|T_D^{(\ell)}}^{(\ell)}, p_{T_D^{(\ell)}|T_D^{(\ell)}}^{(\ell)} \right) \quad (3.24)$$

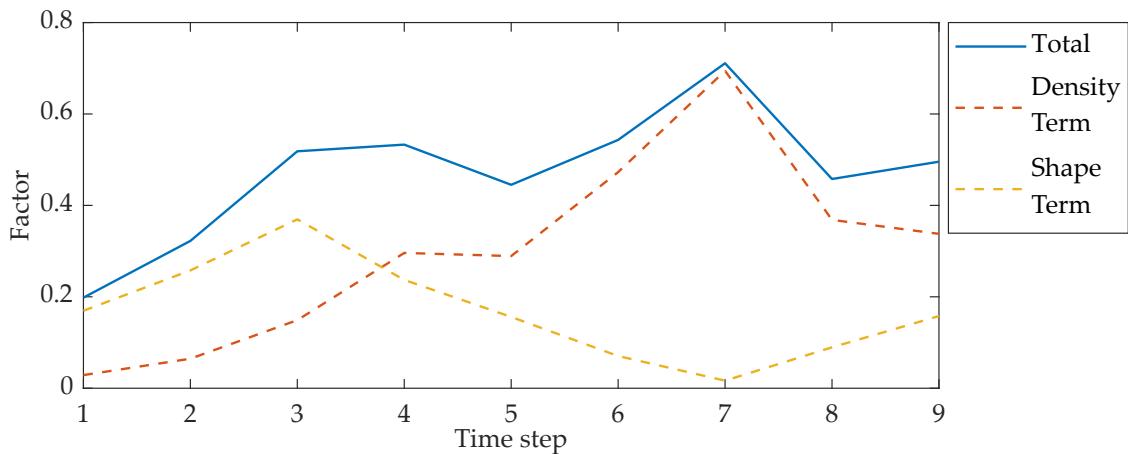
As discussed in section 3.2.4, Gaussian mixtures will normally be generated in the posterior probability distribution for states in the GLMB-to-LMB approximation process, which can in many application cases be approximated by a single Gaussian depending mainly on the dominant component using the mixture reduction algorithm, and thus enables the dealing of the division operation in (3.22) with an extended Rauch-Tung-Striebel smoother [71]. Alternatively, a particle filter can be used as in [28], which requires much more computation yet is assumed to produce limited performance improvement in this case.

### 3.4 Track-Before-Detect State Generation

The robust birth policy to adjust the existence probability of cluster candidates in section 3.2.2 enforces the initialization of newborn objects to be executed only when the cluster candidate is of high quality regarding birthing, which on the other hand may also postpone the confirmation of new tracks whose early states might be ignored. This can especially happen when an object enters the surveillance region from a relatively distant position and approaches the sensor, since the number of measurements from the same



(a) Sequence of measurement sets from an object approaching the sensor.



(b) The adjusting factors for existence probability in the birth model of each time step as in equation (3.10). Note that the adjusting factor will only be calculated before the track is really initialized, and the rest part of this line chart is just to demonstrate its relative level.

Figure 3.10: Example of an object approaching the sensor from a distant position and the corresponding adjusting factors for existence probability regarding birthing. The track is initialized at time step 3 with two missing states.

object usually decreases as the distance to the sensor grows larger and the observation is usually more incomplete, as shown in the example in Fig. 3.10. These missing states cannot be recovered by the backward smoothing either, because the birth time is also postponed accordingly, and no posterior state is generated for smoothing in these early time steps.

To achieve higher completeness for tracks, the backward filtering process is proposed in this thesis to generate track-before-detect estimations based on the unassociated measurements from the forward-backward smoothing process and recovers the missing states starting from the birth time of each track. Note that instead of its original intention for tracking on radar intensity images, we use the term *track-before-detect* to denote the utilization of all recorded LiDAR data in the proposed offline tracking scheme to also produce state estimations before the first detection of an object. This section discusses the principle idea of the backward filter as well as the corresponding modifications

made to the LMB filter that are different from the forward pass.

As discussed above, missing states of this type can only exist in the early stage of tracks, since the initialization of the state (and thus the existence probability adjustment) needs to be calculated only once at the beginning of each track. Also, the consecutive dynamic birth model in section 3.2.2 that requires a pair of measurement clusters will always generate a one-step latency in birthing since the initialization can only be calculated for the latter cluster of the candidate pair in the forward pass. The compensation of such states can then be carried out by a "look-back" operation after the forward-backward smoothing process to possibly incorporate the unassociated measurement clusters into the existing tracks and find these missing states.

After forward-backward smoothing, the cluster associated to an extracted state of track  $\ell$  at time step  $k$  can be represented by  $\hat{\mathcal{U}}_{\hat{\theta}_k(\ell)}(Z_k)$  with  $\hat{\theta}_k(\ell) > 0$ , where  $\hat{\mathcal{U}}_k$  and  $\hat{\theta}_k$  are the partition and the association map in the state extraction of time  $k$ . The set of unassociated data can be removed from the original dataset  $Z = \{Z_k\}_{k=1}^T$  to form a new one composed only of measurements with no specific association, i.e.,

$$Z'_k = Z_k \setminus \left\{ \hat{\mathcal{U}}_{\hat{\theta}_k(\ell)}(Z_k) : \hat{\theta}_k(\ell) > 0 \right\}_{\ell \in \mathbb{L}}. \quad (3.25)$$

Such dataset  $Z' = \{Z'_k\}_{k=1}^T$  corresponds to measurements either from undiscovered states of some tracks or from the clutter (false alarms). To solve the data association problem in the *look-back* operation, another LMB filter is used starting from the last time step  $T$  and runs on  $Z'$  backward in time to search for missing states in each track, as shown in the left part of Fig. 3.1. Simultaneously, states will be extracted and attached to the corresponding tracks from the forward-backward smoothing process to form the final tracking results that are more complete. The LMB filter applied in the backward filtering is mostly similar to the one in the forward pass, except for the following modifications:

- **Noise in Gaussian process** As the contour estimation of an object is relatively accurate after the smoothing process and that the distant measurement points are less reliable, the process noise variance for Gaussian process is decreased while the observation noise variance is increased, which means the filter tends to trust more in the previous estimation from the forward pass and less in the unassociated measurements.
- **Survival and detection probability** The survival probability  $p_S$  is set higher compared to the value in forward-backward smoothing to increase the existence probability of the predicted tracks as in (3.8), while the detection probability  $p_D$  is set lower to be more tolerant of object occlusion as in (3.11). The reason behind this adjustment will be investigated and analyzed later in the performance evaluation chapter.
- **Minimum point amount in cluster** A threshold for the minimum amount of points in a cluster is still used in measurement set partitioning as in the forward

pass to distinguish meaningful clusters from clutter, yet it is reduced to allow smaller cluster to come into the update step so that possibly more unassociated data will be utilized.

- **Birth policy** Instead of performing the consecutive measurement-based adaptive birth as in the forward pass (see section 3.2.2), the state of track  $\ell$  is initialized automatically in the backward filtering process at its original birth time  $k_B^{(\ell)}$  with identical probability distribution for contour and reversed kinematics as from smoothing, i.e.,

$$\begin{aligned}\tilde{\pi}_B^{(\ell)} &= \left( r_{k_B^{(\ell)}|T}^{(\ell)}, \tilde{p}_{k_B^{(\ell)}|T}^{(\ell)} \right), \\ \tilde{p}_{k_B^{(\ell)}|T}^{(\ell)} &\sim \mathcal{N} \left( A_R \underline{x}_{k_B^{(\ell)}|T}^{(\ell)}, \Sigma_{k_B^{(\ell)}|T}^{(\ell)} \right),\end{aligned}\tag{3.26}$$

where  $\underline{x}_{k_B^{(\ell)}|T}^{(\ell)}$  and  $\Sigma_{k_B^{(\ell)}|T}^{(\ell)}$  denote the mean and the covariance matrix of track  $\ell$  at  $k_B^{(\ell)}$  after smoothing, and the matrix  $A_R$  reverses the direction of velocity, turn rate and acceleration (if applicable, depending on the selected motion model) of the extended state vector while keeping the basis vector unchanged. For example, with the CTRA model  $\underline{x}^k = [x \ y \ \psi \ v \ \omega \ a]^\top$  in the extended state vector, the matrix is given by

$$A_R = \begin{bmatrix} 1 & & & \dots & & 0 \\ & 1 & & & & \\ & & 1 & & & \\ \vdots & & & -1 & & \vdots \\ & & & & -1 & \\ & & & & & -1 \\ 0 & & & \dots & & I_{N^b} \end{bmatrix}\tag{3.27}$$

The existence probability and distribution of each track are then propagated backward by the backward filter in the same *Predict-Update* manner as described in section 3.2 until being pruned due to low existence probability, the time step of which then corresponds (or at least is closer) to its true birth in the original dataset.



# 4 Performance Evaluation

This chapter discusses the previously introduced track-before-detect smoothing approach for multiple extended objects from a practical perspective by examining its performance in various aspects. To achieve this goal, the test perspectives and the corresponding metrics are given at first to clarify the evaluation processes, followed by the illustration and analysis for each test scenario. The proposed approach is evaluated with both simulational and real-world data, so that possibly more typical scenarios can be investigated.

## 4.1 Test Perspectives and Metrics

The performance perspectives required for evaluation of the approach can be roughly categorized into two classes:

- The performance in estimating the kinematic states and the contour shape of each individual object under the multi-object tracking assumption. This class mainly involves the performance improvement of smoothing compared to filtering. Since the tracks are independent of each other in the form of an LMB and are only correlated in the update process due to data association uncertainties, this class of performance can be taken as similar to the single object tracking problem except for the dynamic object birth, the object occlusion and the object death that can influence the number of objects. Accordingly, an examination of performance for each individual track would already be sufficient for this class, which is investigated in section 4.2.2 and section 4.3.
- The performance in estimating the overall cardinality of the multi-object state at each time step. This class mainly comprises the correctness of solving the data association problem and yielding the correct number of objects, and emphasizes the effectiveness of track management, missing state recovery and cardinality correction in the post-processing phase. While the state estimation is still evaluated for each object, it is combined with cardinality measures to further demonstrate the overall performance of multi-object tracking. The evaluations of this class are presented in section 4.2.3.

As measures for these two classes of performance evaluation, the following metrics are used in this chapter for different purposes:

- The Intersection over Union (IoU) ratio shows the extent to which two areas overlap, and is applied in this thesis mainly to demonstrate the quality of the

contour estimations for extended objects while also capturing the errors in the position  $(x, y)$  and the orientation  $\psi$ . For two areas  $A_1$  and  $A_2$ , the IoU ratio is defined as:

$$\text{IoU} = \frac{\text{area}(A_1 \cap A_2)}{\text{area}(A_1 \cup A_2)} \in [0, 1] \quad (4.1)$$

where  $\text{IoU} = 1$  represents a perfectly estimated contour w.r.t. the reference, and  $\text{IoU} = 0$  represents no overlap between these two at all.

- The Optimal SubPattern Assignment (OSPA) metric proposed in [72] is a performance measure especially for multiple object tracking, taking into account the error in both individual states and the cardinality estimation. Based on a cut-off parameter  $c$ , the distance between two individual states  $\underline{x}_1$  and  $\underline{x}_2$  is defined as  $d^{(c)}(\underline{x}_1, \underline{x}_2) = \min(c, \|\underline{x}_1 - \underline{x}_2\|)$  with  $\|\underline{x}_1 - \underline{x}_2\|$  as the Euclidean distance. The overall OSPA distance between two multi-object state sets  $X, Y : |X| \geq |Y|$  can then be specified with an order parameter  $p$  as:

$$\bar{d}_p^{(c)}(X, Y) := \left( \frac{1}{n} \left( \min_{\theta \in \Theta} \sum_{i=1}^{|X|} d^{(c)}(\underline{x}_i, \underline{y}_{\theta(i)})^p + c^p (|X| - |Y|) \right) \right)^{1/p} \quad (4.2)$$

and for  $|X| < |Y|$ ,  $\bar{d}_p^{(c)}(X, Y) := \bar{d}_p^{(c)}(Y, X)$ , i.e., the sum of all pair distances under the optimal assignment.

The cut-off parameter  $c$  determines the weighting of how the metric penalizes the cardinality errors as opposed to the localization errors. For  $p = 1$ , it is exactly the penalty given to any false or missing state estimation. It thus represents the threshold at which we no longer distinguish whether two states are paired together or whether one of them remains unassigned while the other can make an exact match. If they were any closer together, we would prefer pairing the two points; if they were further apart, we would prefer to say that they are unrelated.

The order parameter  $p$  controls the weights assigned to outliers. As  $p$  increases and  $c$  remains fixed, the OSPA metric becomes less tolerant to estimations that are not close to any objects of the ground-truth. Note that this effect is somewhat mitigated by the fact that the distances between points are cut off at  $c$ , which is also the penalty that a point gets when it is deemed “unassignable”.

## 4.2 Simulations

In this section, various perspectives of the proposed tracking scheme is tested in the corresponding simulation scenarios after the simulation setup being introduced. For the single object simulations, different motion models are compared according to the kinematic and contour estimations of the tracking results for both straight and curvilinear trajectories; then the filtered and the smoothed results of tracking a single maneuvering object is compared to illustrate the performance improvement in smoothing; the influence

of selecting different length scales in the Gaussian process for contour estimation is also illustrated and analyzed. Combined together, these simulations show the performance of the proposed scheme in estimating the states of each individual extended object. For the multi-object simulation, objects performing typical traffic behaviors like steering, following, paralleling, etc. are tracked to further illustrate the method's performance in arranging tracks under complicated circumstances and generating track-before-detect estimations.

### 4.2.1 Simulation Setup

In the simulations, objects are abstracted to be rectangles of various geometries, and data points are generated as if coming from a one-layer LiDAR positioned at  $(0, 0)$  with  $360^\circ$  field of view, horizontal angular resolution  $D = 1/6 \text{ deg}$  and sampling interval  $\Delta t = 0.1\text{s}$ . Points are only generated on the visible sides of the objects to the sensors, and possible object occlusion is also taken into account.

The surveillance region is the square area  $[-80, 80] \times [-80, 80]$  (units in meter), corresponding to the typical sensing range of a LiDAR, and the clutter is assumed to be uniformly distributed, whose number follows the Poisson distribution with  $\lambda = 15$ . Process noises for acceleration and turn rate are assumed to be zero-mean Gaussian with  $\sigma_a = 50 \frac{\text{m}}{\text{s}^3} \cdot \Delta t$  and  $\sigma_\omega = 200 \frac{\text{deg}}{\text{s}^2} \cdot \Delta t$ , which resemble the normal kinematics of road vehicles. Similarly,  $\sigma_M = 5 \text{ cm}$  is set as measurement noise for each single LiDAR point. The survival and detection probabilities are set as  $p_S = 0.99$  and  $p_D = 0.9$  for both forward and backward passes at all time steps.

### 4.2.2 Single Object Simulations

#### Comparison of Motion Models

As stated in section 2.2, the choice of motion model in the tracking process decides the type of motion that can be tracked, and is crucial to obtaining estimations of high quality. In this section of simulation, the proposed tracking scheme with different embedded motion models is tested in two scenarios, where only the final results (after backward filtering) are plotted in the figures.

In the first scenario, the trajectory of the object is linear with changing velocity, so the comparison is performed between the Constant Velocity (CV) model and the Constant Turn Rate Acceleration (CTRA) model to demonstrate the advantage in directly modeling the acceleration, as in Fig. 4.1. The object starts with constant velocity, accelerates with an acceleration of  $2\text{m/s}^2$  between time steps 31 and 60 (3 seconds), and then keeps the velocity constant again. As can be observed in the position error and the velocity estimation, both models can yield relative precise results when the velocity is constant; however, the CV model fails to adapt to the velocity change as the object begins to accelerate (and actually loses the object from time step 67). Meanwhile, the CTRA model still keeps up with the changing velocity and further provides precise estimations, since

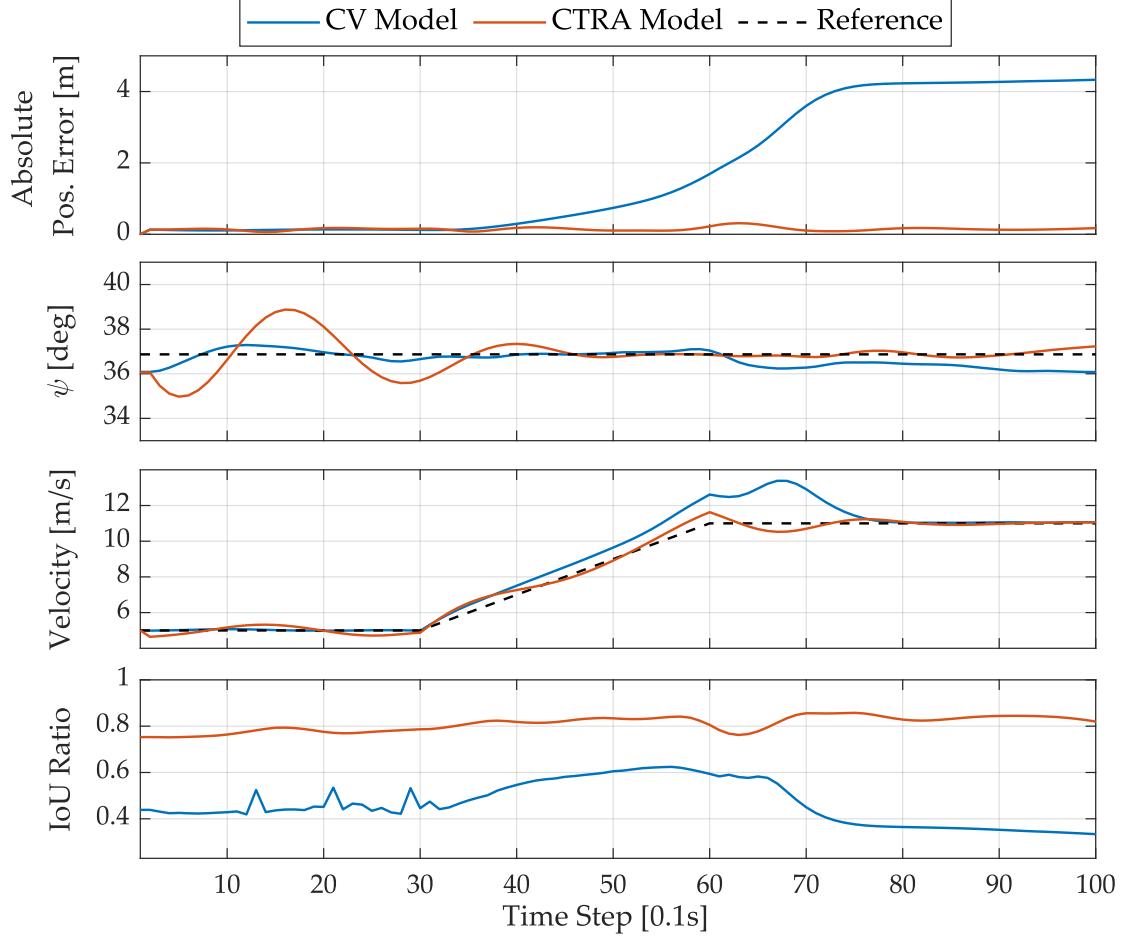


Figure 4.1: Simulation results of tracking an extended object on a straight trajectory with the CV and CTRA motion models.

it explicitly models the acceleration in the state vector and also estimates its variation over time. This improvement in performance can also be observed in the IoU ratio that is influenced by both the kinematic and the contour estimations. The estimated orientation given by the CTRA model fluctuates at the beginning since the model also considers a possible turn rate, but quickly approaches the true value after several time steps.

In the second scenario, the object performs accelerations with additional turn rate changes, as illustrated in Fig. 4.2. Unlike the first scenario, the CV model is not involved here due to the curvilinear trajectory, and the comparison between the Constant Turn Rate Velocity (CTRV) model and the CTRA model is examined in various aspects. Although the results of both models are closely bounded to the reference values, the CTRA model shows its superiority over the CTRV model in that it responds faster to velocity changes and yields slightly better orientation estimations, and thus smaller position error and higher IoU ratio. Note that at time step 40, the CTRV model fails to follow the velocity change as the object decelerates and only approaches the true value

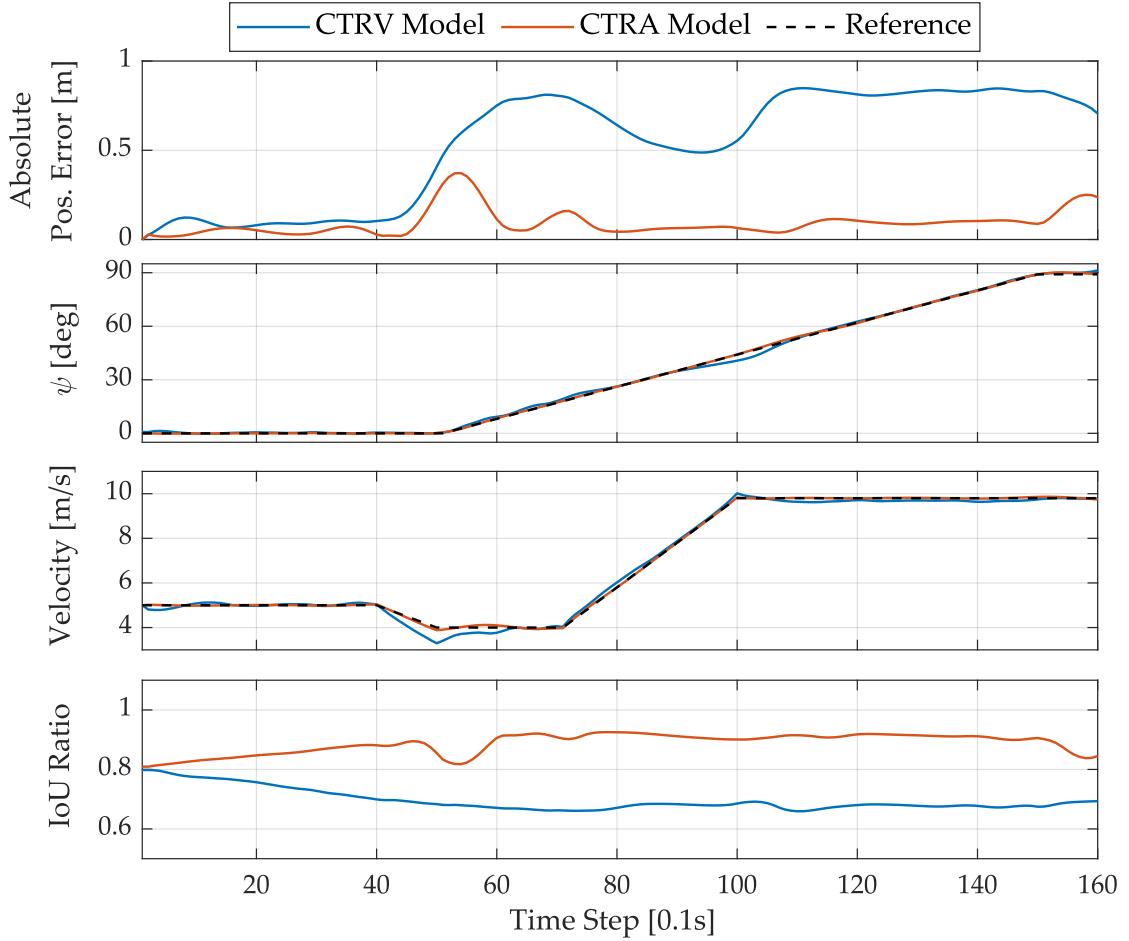


Figure 4.2: Simulation results of tracking a maneuvering extended object with the CTRV and CTRA motion models.

at around time step 60, resulting in larger position error in between and afterwards.

By comparing the tracking results of both scenarios, it can be concluded that although all three motion models can generate relatively precise kinematic estimation when the velocity remains constant, the CTRA model is the most practical model to track an object whose the velocity changes over time due to its explicit modeling of the acceleration, as analyzed in section 2.2.3. Note that one disadvantage of the CTRA model is that the estimation of the initial acceleration requires at least three clusters from different time steps and is not robust with a short sampling interval, which can be alternatively solved by setting all initial acceleration as zero instead and set a large initial variance for it. In this way, the acceleration will be quickly corrected by measurement updates in the filtering process. From this point on, the CTRA model and the zero initialization for the acceleration will be applied as standard motion model in all following simulations and experiments.

### Comparison of Filtered and Smoothed Results

In this section, the filtered and the smoothed tracking results are compared to demonstrate the influence of track smoothing. Fig. 4.3 shows the simulated test scenario as well as the velocity and orientation of the object over time. The object starts with constant velocity and turn rate, and accelerates with changing acceleration while also performing steering simultaneously.

The simulation results are shown in Fig. 4.4. As can be noticed from Fig. 4.4c, the contour estimation in the forward pass has been largely improved after time step 80 due to the arrival of new information from the other side of the contour that just became visible to the sensor (i.e., after the object bypasses the sensor). By comparing Fig. 4.4a and 4.4c, fluctuations of IoU during the forward pass can be partly explained as side effect of estimation errors in kinematics, and are mostly reduced in the final results with improved contour and position estimation.

After the filtering process, the improved contour with information from all sides of the object is propagated backward in smoothing and thus results in more complete contour estimation in the first 80 steps. As a consequence, the errors in position and velocity estimation have also been reduced as illustrated in Fig. 4.4a and 4.4b. Furthermore, it can be noticed that the one-step latency at the beginning of the forward pass caused by the adaptive birth (see section 3.2.2 and analysis in section 3.4) has been corrected after applying track-before-detect generation.

To test the flexibility of estimating free forms, no prior knowledge of the object contour is assumed except for smoothness and closure, thus no advanced kernel function (e.g., symmetric, conservative kernels or kernels with explicit basis function [43]) is utilized in the Gaussian process to constrain the approximation of the radial function,

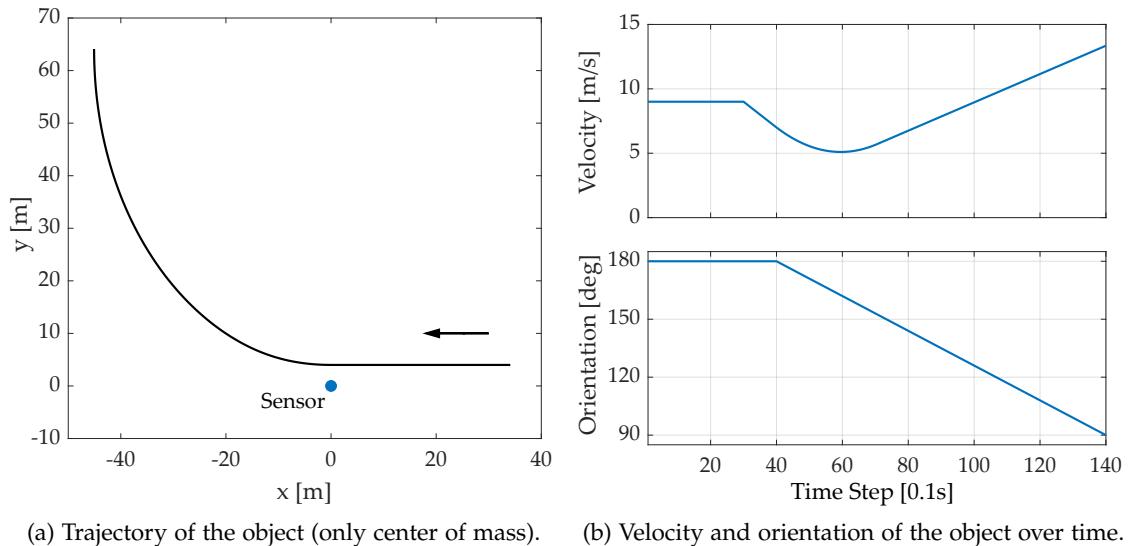


Figure 4.3: Test scenario of tracking a maneuvering extended object.

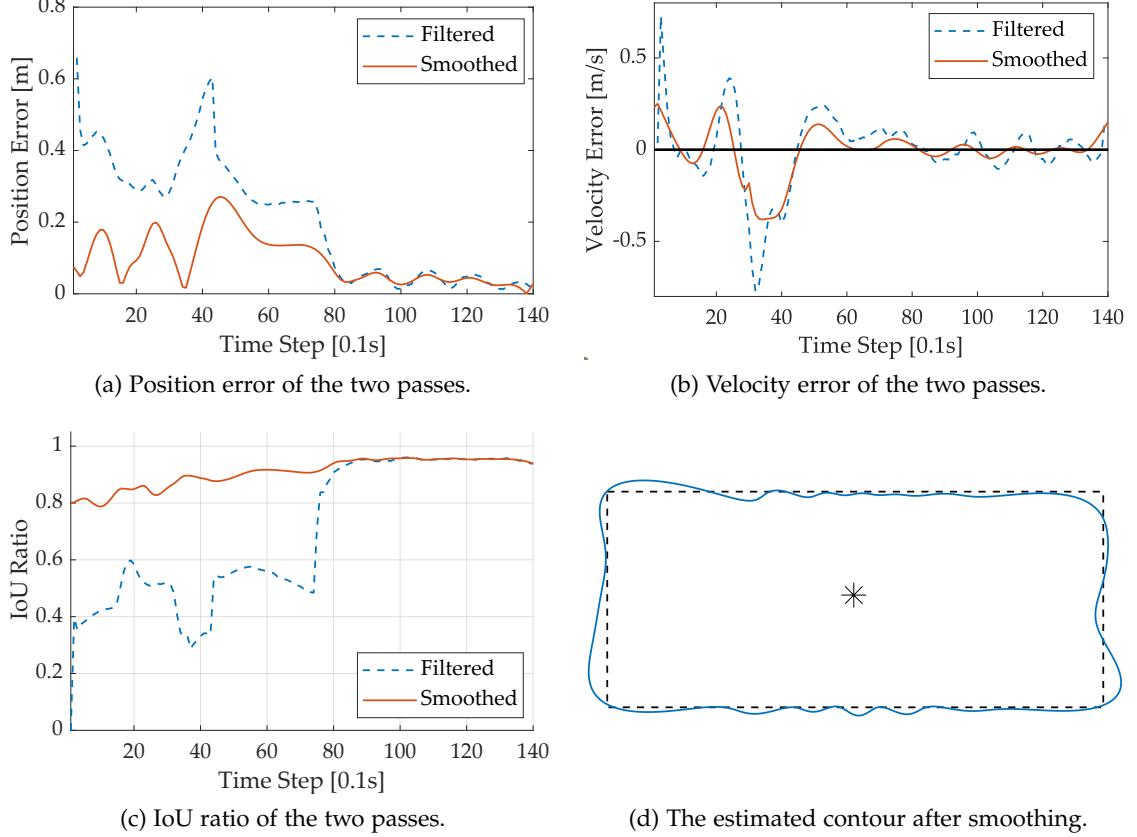


Figure 4.4: Simulation results of tracking a maneuvering extended object and the comparison between filtering and smoothing.

resulting in slightly irregular contour estimations as in Fig. 4.4d. Nonetheless, for some specific applications where assumptions can be made for the geometry, the symmetry or the completeness of the contour, certain kernel functions can notably enhance the performance of the contour estimation [43, 44].

### Influence of Length Scale Selection

In this section, the influence of length scale selection in the kernel function (2.44) of Gaussian process is investigated by reusing the test scenario in Fig. 4.3 and testing with different length scale values, and the results are illustrated in Fig. 4.5. As can be observed in the test results, the contour estimation can vary largely according to the length scale selection.

In Fig. 4.5a, the length scale is selected to be very small, indicating that the covariance between an angle of the basis points and its neighboring angles would be correspondingly small. Thus, the influence of a measurement point (in the form of a polar coordinate pair w.r.t. the reference point) would hardly reach its nearest basis

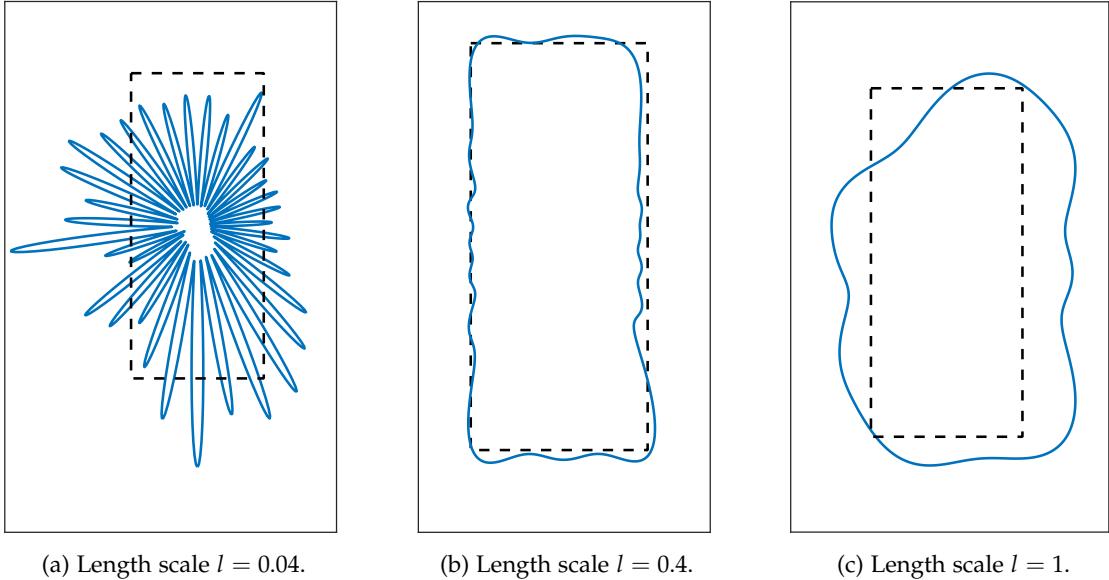


Figure 4.5: Influence of length scale selection in the kernel function on tracking a single maneuvering extended object. Solid lines are the final contour estimations, while the dashed lines are the reference contours.

point in the update process unless the angular distance is close to zero. This result in not only the limited improvement of update with each measurement point, but also the unstable prediction of the radial function based on the basis vector when plotting the result, as can be easily observed in Fig. 4.5a. In the extreme situation,  $l \rightarrow 0$  will result in all zero values for the whole basis vector unless a basis point exactly equals to the local polar angle of some measurement point.

On the other hand, a too large value for the length scale will yield large covariance among angles, which only slowly decreases with the growth of the angular distance. The influence range of a measurement point would be too wide in the update, which could change the function value of more basis points even if it is not close to the current measurement point. In such manner, details of the contour would be lost, as shown in Fig. 4.5c. In the extreme situation where  $l \rightarrow \infty$ , the estimated contour will take the form of a perfect circle since the covariance among all angles will be almost equal.

An appropriate length scale in the kernel function will yield contour estimation of high quality, as shown in Fig. 4.5b. The selection of this value depends on the geometry and form of the object contour, and affects the covariance between angles jointly with the amplitude parameter  $\sigma_F$  as discussed in section 2.4.2.

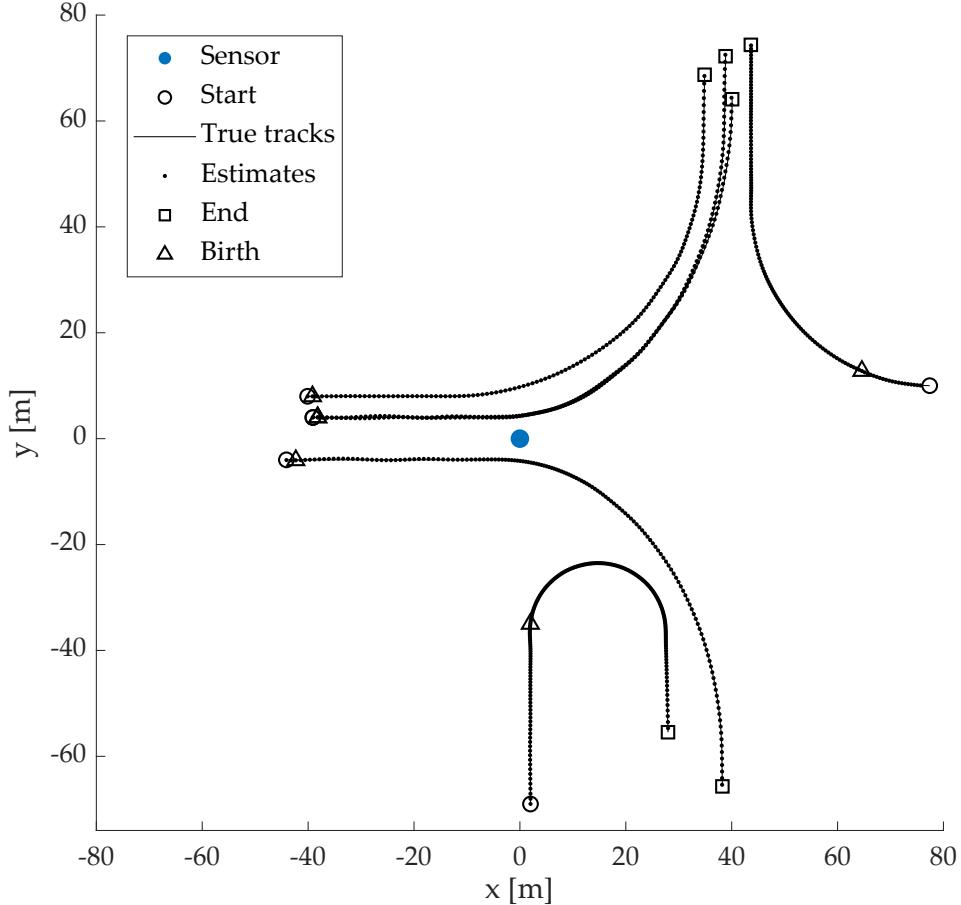


Figure 4.6: True and estimated trajectories for tracking multiple extended objects (only the center of mass is shown for the position estimations).

#### 4.2.3 Multiple Object Simulation

In this section, the proposed scheme is tested in a multi-object scenario to demonstrate its performance in simultaneously estimating multiple tracks and generating track-before-detect estimations by post-processing the unassigned measurements. Six objects with various geometries and birth/death intervals are present to simulate possible traffic behaviors at an intersection, whose true and estimated trajectories are depicted in Fig. 4.6. Using the IoU ratio and the OSPA metric, the proposed track-before-detect method (Smoother with Robust Birth and Backward Filter, S-RB-BF) is compared against naive LMB smoother for extended objects and Smoother with Robust Birth (S-RB) in Table 4.1 and Fig. 4.7.

In Fig. 4.6, the original birth times for objects estimated in the forward pass are denoted as triangles on each individual track, and some missing states can be observed as objects enter the scene (especially for the objects entering from bottom and right due to limited information on the shape geometry and too small number of measurement

Table 4.1: Average IoU ratios over time for each track

Tracks	Smoother	S-RB	S-RB-BF
1	0.8387	0.8387	<u>0.8445</u>
2	0.8863	0.8863	<u>0.8917</u>
3	0.4837	0.8833	<u>0.8939</u>
4	0.8885	0.8885	<u>0.8946</u>
5	0.6744	0.6512	<u>0.8675</u>
6	0.7333	0.6424	<u>0.7490</u>

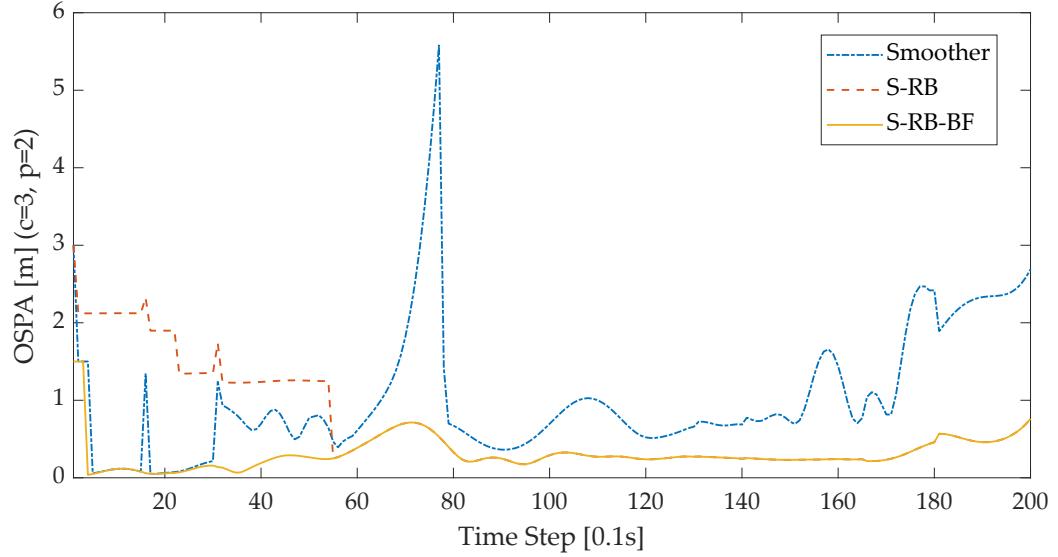
points, respectively). After the backward pass, the proposed backward filter recovers these missing states and finally generates estimations which accurately follow the true tracks with a low cardinality error.

Comparing the OSPA errors of the naive LMB smoother and S-RB in Fig. 4.7a, it can be observed that although the initial OSPA errors are low for the smoother without robust birth, track 3 is quickly lost due to a poor contour initialization (cf. the initial IoU of track 3 in Fig. 4.7b). In contrast, missing states in the early steps of S-RB cause high OSPA errors (e.g., for tracks 3 and 5), which quickly decrease after all objects are tracked. The OSPA errors in the later steps are identical between S-RB and S-RB-BF since the backward filter starts for each track at its original birth time estimated in the forward pass, while S-RB-BF effectively recovers most missing states in the early steps and thus largely reduced the OSPA errors in this period. Note especially that the one-step latency from the adaptive birth is canceled out by the backward filter for all tracks except for track 5, which generates too few measurement points in the first several steps and cannot be recovered even by the backward filter.

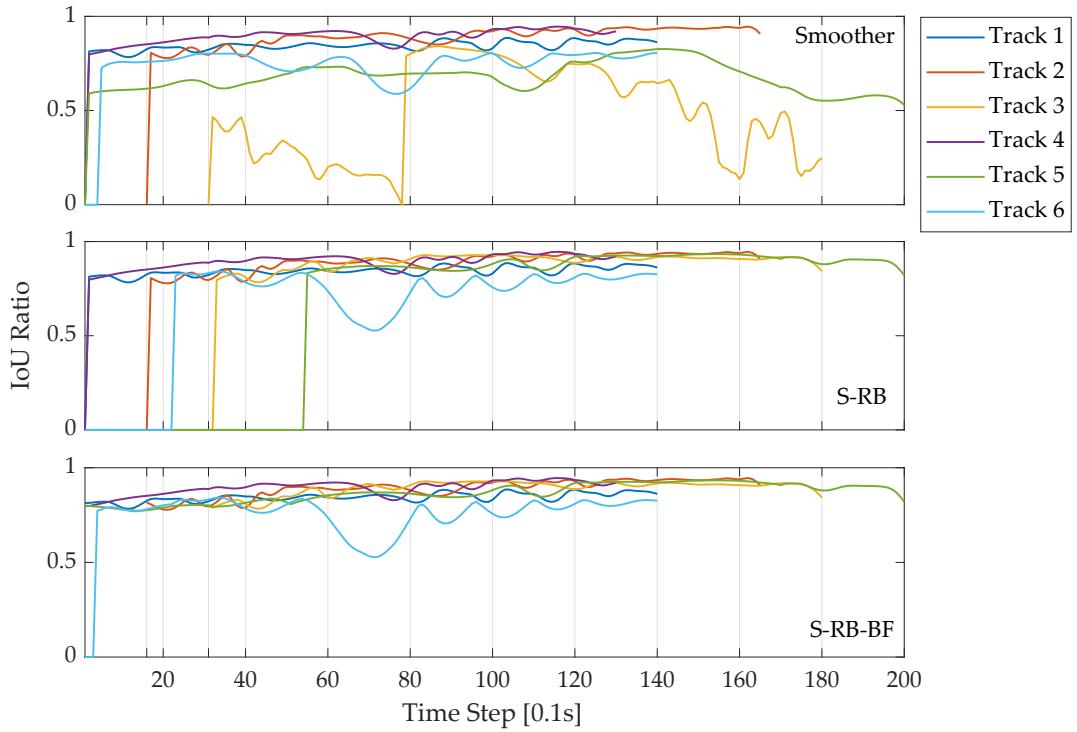
The IoU ratios of the contour estimations with different methods are shown in Fig. 4.7b for each track over time, and are averaged over time and shown in Table 4.1. Comparing the results of the naive smoother and S-RB, it can be seen that the robust birth policy yields much better contour estimations for all tracks on the initialization, yet also much postponed birth time for tracks 5 and 6 (entering from the bottom and the right borders in Fig. 4.6). This initial cardinality error leads to even lower average IoU for these two tracks in Table 4.1 since IoU equals zero for any misdected object. Comparing the results of S-RB and S-RB-BF, it can be easily observed that S-RB-BF recovers almost all missing states from the robust birth policy while also preserves the stable estimations of the following steps, and produces overall better results for all tracks (comparing columns 2 and 4).

### 4.3 Real-world Experiments

In this section, the proposed method is applied to data collected from a test drive to illustrate its capability in handling the real-world scenario and estimating more realistic trajectories and motion. The first subsection provides details for the data acquisition



(a) OSPA error of estimations from the three methods.



(b) IoU ratio for individual tracks from the three methods.

Figure 4.7: OSPA error and IoU ratio of the naive smoother, S-RB and S-RB-BF.

process and the hardware platform; the results of tracking an extended object from the collected data are presented afterwards and compared to the reference results from the DGPS to evaluate its performance; at last, the influence of value selection for the

detection probability  $p_D$  on the LMB filter is investigated and analyzed, which is crucial for the tracker to handle occlusion and to yield correct cardinality estimation.

### 4.3.1 Data Acquisition

The real-world scenario is obtained by a test drive recorded on the BMW testing ground, as shown in the bird-eye view in Fig. 4.8. As the object to be tracked, a BMW 740Li vehicle with a known contour shape is considered as the target vehicle from which the LiDAR measurements are generated. In this scenario with a total duration of 40 seconds, the target vehicle maneuvers within the distance range between 8m and 60m to the ego-vehicle. The speed of the target vehicle varies over time between  $10\text{km}/\text{h}$  and  $30\text{km}/\text{h}$  with a maximal acceleration of  $2\text{m}/\text{s}^2$ . Two  $180^\circ$  steerings are performed by the target vehicle, both at the most distant positions to the ego-vehicle, with a maximal turn rate of  $46^\circ/\text{s}$  and a maximal angular acceleration of  $5^\circ/\text{s}^2$ .

In this scenario, the acquired data is a sequence of point cloud frames collected by a Hesai Pandar LiDAR sensor mounted on the roof structure of the ego-vehicle, which is marked with the red rectangle and zoomed in Fig. 4.9. The sensor has  $360^\circ$  field of view, 40 vertical channels, a horizontal angular resolution of  $0.2^\circ$  per layer, and the sampling interval  $\Delta t = 0.1\text{s}$ . All collected measurement points are projected onto the X-Y plane since the implementation of the proposed method only tackles 2D estimations, and possible contour points (points visible to the sensor on the 2D plane) are extracted as described in [73].

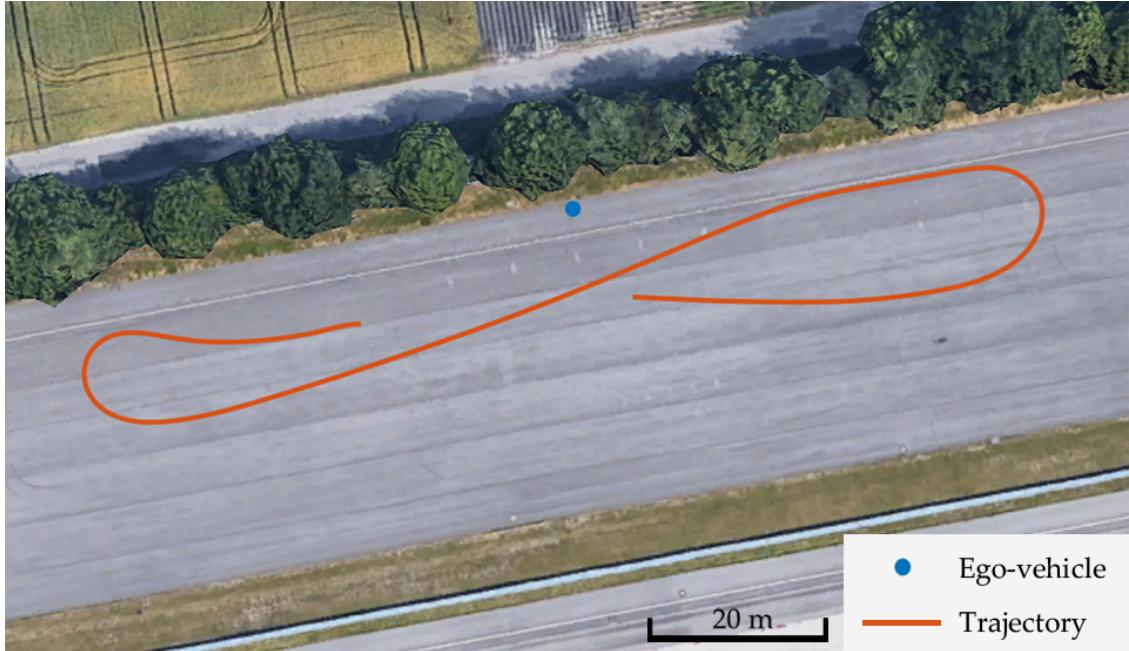


Figure 4.8: Bird eye view of the testing ground and the trajectory of the target vehicle.



Figure 4.9: The test vehicle used for data acquisition and the LiDAR sensor setup on the roof structure.

With regard to the evaluation of the proposed tracking scheme, also the reference data describing the object states are required. To this end, the RT-Range system from Oxford Technical Solutions Ltd has been employed to obtain the relative motion data of the target vehicle w.r.t. the ego-vehicle. The installation includes a RT3000 measurement device for both the ego-vehicle and the target vehicle to perform DGPS aided inertial navigation. By exploiting the real-time kinematic positioning principle, a localization precision of up to 1cm can be achieved [74]. The relative positions of the target vehicle are compared offline with the DGPS-recordings of both vehicles for ego-motion compensation. The RT-Range target equipment has been installed on the target vehicle, from which the information about the relative heading, the 2D range and the relative 2D velocities will be extracted for the evaluations in the next section.

### 4.3.2 Test Results

The proposed tracking scheme is applied to the aforementioned test scenario with again the CTRA motion model and some main parameters given in Table 4.2. As discussed in section 2.2, the initial variance for acceleration is set to be large enough to allow a zero initialization. Also, as the sampling interval of the sensor is short, the initial guess for the turn rate may also deviate considerably due to the accumulated error in the orientation estimations of the two consecutive clusters in birthing. Thus, the initial

Table 4.2: Parameterization for the test on real-world data

Parameter	Value	Description
$\sigma_a$	$50m/s^3 \cdot \Delta t$	Process noise variance for acceleration
$\sigma_\omega$	$100^\circ/s^2 \cdot \Delta t$	Process noise variance for turn rate
$\sigma_{a,init}$	$100m/s^3 \cdot \Delta t$	Initial variance for acceleration after birth
$\sigma_{\omega,init}$	$300^\circ/s^2 \cdot \Delta t$	Initial variance for turn rate after birth
$p_S$	0.9	Survival probability of an individual track for LMB
$p_D$	0.6	Detection probability of an individual track for LMB
$\sigma_F$	0.7	Amplitude in kernel function for Gaussian process
$l$	0.4	Length scale in kernel function for Gaussian process

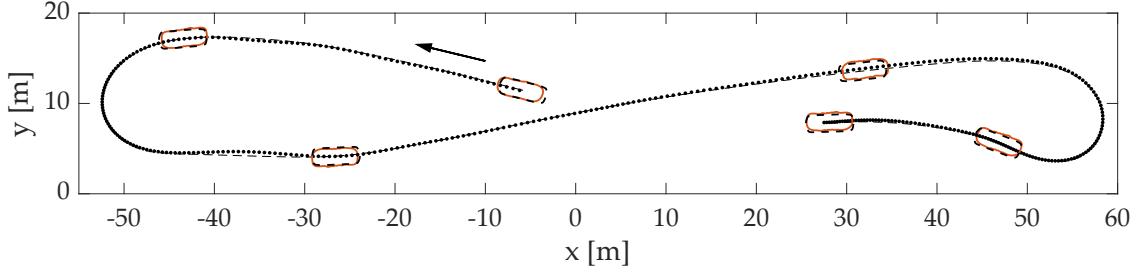
variance for the turn rate is also set relatively large to address this problem.

The DGPS-recorded and the estimated object trajectories are shown in Fig. 4.10a where each dot denotes the center of mass of an estimated state, and the corresponding contour estimations are shown for times steps  $k \in \{1, 60, 150, 240, 340, 390\}$ . As can be seen from the results, the estimated trajectory is tightly bounded around the DGPS-recorded positions, and the contour estimation also yields a low deviation from the true contour, demonstrating the feasibility of the proposed method in practical cases.

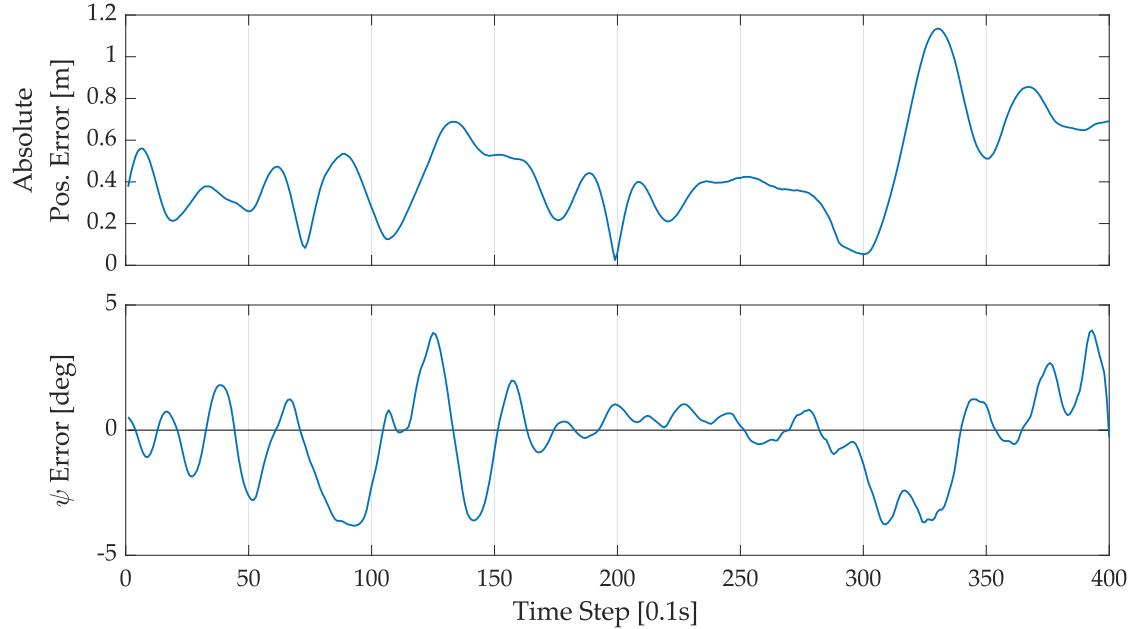
The position and the orientation errors are shown in Fig. 4.10b over time, which are also kept at relatively low level as can be observed. The largest estimation errors occur when the target vehicle is at the most distant positions to the sensor, corresponding to the time steps 80 – 130 and 290 – 330 where the amounts of points in the cluster are relatively small and the quality of measurement update is limited. Also, the geometrical distribution of the points over the contour can influence the accuracy of the estimations, especially when all points in the cluster are coming from the same side of the object contour (i.e., the I-shape in Fig. 3.4a), which is the case when the target vehicle bypasses the ego-vehicle or moves roughly towards the direction pointing to it. In such situation, little information on the kinematics in the orthogonal direction is provided by the cluster, which can possibly result in inaccurate estimation when the velocity changes (e.g., the large errors at time steps 125 and 330).

### 4.3.3 Influence of Detection Probability

In this section, the influence of a crucial parameter to the LMB filter, the detection probability  $p_D$ , is investigated. According to different value selection for this parameter, the aspect of the filter that is mainly affected would be the state extraction (and thus the cardinality estimation, see section 3.2.5) after the GLMB update and the LMB approximation of each time step. The test is again carried out on the real-world data with different  $p_D$  values, and the number of points of the object cluster as well as the estimated cardinality given by the filter and the smoother are shown for a certain time interval in Fig. 4.11.



(a) Trajectory and contour estimations generated by the proposed method (points and solid lines), and DGPS-recorded trajectory with true contours (dashed lines).



(b) Position and orientation errors of the two passes.

Figure 4.10: True and estimated trajectories and contours, and the corresponding errors of the real-world experiment.

As the target vehicle drives further from the ego-vehicle and the amount of measurement points decreases crossing below some threshold, the cluster would no longer be regarded as valid for measurement update (e.g., for time steps 285 – 290 and 296 – 304), and the filter tends to consider the current state as misdetected. In this kind of situation, the predicted state from the last time step will be directly propagated forward without update, and the likelihood of the current individual state is decided by the second term in (3.12), which in turn influences the multiple object likelihood (3.11), the corresponding hypothesis weights of the posterior GLMB (3.15), and finally the existence probability of the corresponding posterior LMB track (3.16) at the current time step. It can be observed from the figure that some states are not extracted for both filters with higher and lower detection probabilities due to too low existence probability of the LMB track. However,

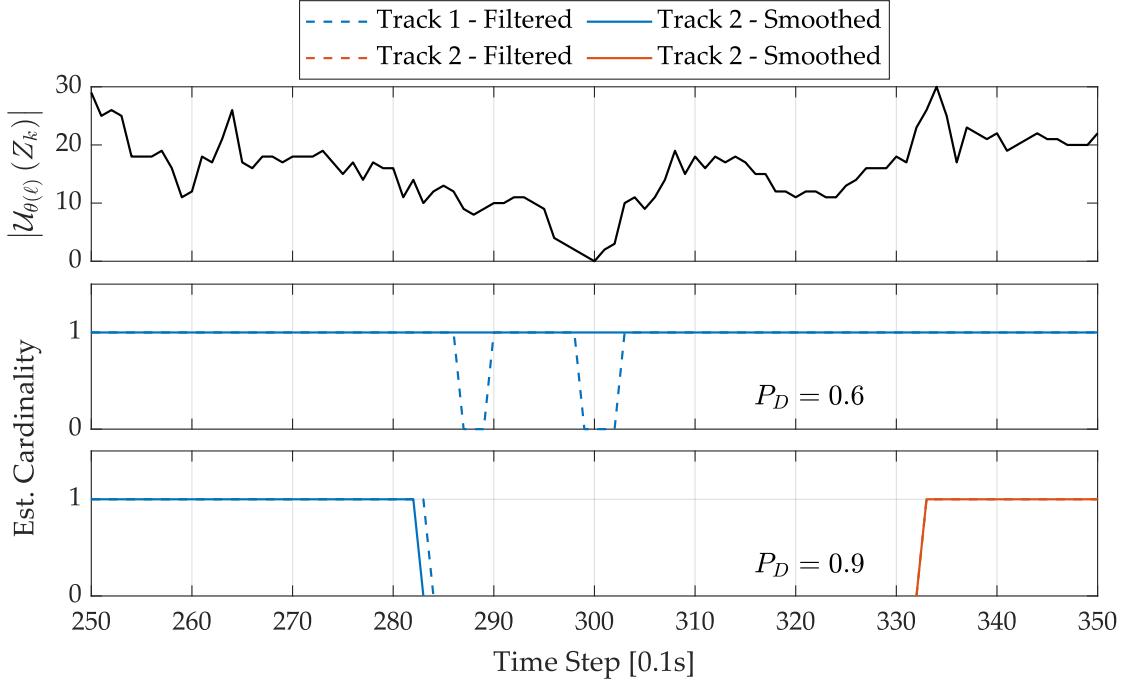


Figure 4.11: Number of points of the object cluster and the cardinality estimation with different  $p_D$  in a certain interval.

less cardinality error occurs for the lower detection probability ( $p_D = 0.6$ ) than the higher ( $p_D = 0.9$ ), which can be explained by the higher single object likelihood (thus higher existence probability of the LMB state) on misdetection.

Furthermore, the cardinality error for  $p_D = 0.6$  can be recovered by the smoother since the existence probability of each state has been corrected in smoothing, whereas it is not recovered when the detection probability is high. The reason behind is that as the object is regarded as misdetected for several continuous steps, the existence probability of the state drops quickly to a very low level due to the multiplication with the likelihood term  $1 - p_D$  at each step, and the track will be pruned when the existence probability becomes too small (threshold set as  $10^{-5}$  in this scenario) at some point, corresponding to object death considered by the filter. Since smoothing starts for each track from its last time step before death, the recovery for the subsequent states becomes impossible. Instead, the filter regards the clusters from this object as part of the clutter until they fulfill the conditions for birthing at some point, where a new track is established independent of the previous track (shown in Fig. 4.11 as track 2).

A straightforward way of understanding the detection probability is to consider it from the perspective of the filter: it is the filter's assumption on the presence of an individual object at all time steps. When a higher value is set for  $p_D$ , it means that the filter expects each object to be present almost all the time, and larger punishment will be exerted on the existence probability on misdetection. This applies also to the survival

probability  $p_S$ , which influences the prediction step (see (3.8) in section 3.2) in a similar way. The appropriate values for these parameters depend on the specific scenario and application, and should be select carefully to avoid broken tracks as discussed above. Note that although set as identical in this thesis for all objects at all time steps in the same scenario, the LMB filter actually allows the detection and survival probabilities to vary depending on the states and labels, which provides great flexibility of selecting different proper values for them according to individual conditions of different objects.



# 5 Conclusion and Perspective

After the mathematical derivation and performance evaluation of the proposed method in the previous chapters, this chapter summarizes the work presented in this thesis, draws conclusions from the investigation and discussions, and outlooks perspectives that can still be implemented or improved in the future to further enhance the performance of the proposed method.

## 5.1 Conclusion

The main objective of this thesis is to generate more accurate tracking estimates for multiple extended objects from LiDAR-recordings, so that they can be utilized as reference data to validate and evaluate the functionalities of ADS. Although the Finite Set Statistics and the star-convex shape model have both been investigated in the literature for tracking multiple extended objects, few researches exist focusing on the offline post-processing of data to improve the quality of tracking and to solve the conflict between completeness and robustness when processing real-world data.

In this thesis, a two-stage tracking scheme has been developed and investigated to contribute to this field. In the first stage, a normal LMB filter is applied with the integration of the robust birth policy to control the quality of initial estimations for newborn objects, and the estimated tracks are smoothed and propagated backward to enhance the quality of the estimations for kinematics and contour shapes; in the second stage, a backward filter starts from the last time step, processes the unassociated data in the first stage and recovers possibly missing states for each track obtained from the robust birth policy. From this concept, all involved processing steps have been explained in detail and moreover, a prototype has been implemented for verification. The performance evaluation have been carried out for both single and multiple object scenarios, where the first class mainly investigates the performance in estimating the kinematic state and the contour shape of each individual object, and the second class focuses on cardinality estimation and state recovery.

From the experiments on single object scenarios, it follows that the CTRA motion model yields the best results for objects with changing velocities on both straight and curvilinear trajectories, since it explicitly models the acceleration and complies with the motion of maneuvering objects. Furthermore, it can be concluded that the smoother can largely improve the estimations of both kinematics and contour shapes, as it integrates information from future time steps and propagates a more complete contour shape backward to the earlier steps. Besides, the influence of two essential parameters has

been investigated. Simulations have shown that the length scale in the kernel function of Gaussian process controls the influence range of each measurement point on the basis vector in the update step, while the detection probability of the LMB filter adjusts the filter's assumption on the detection of each individual object at all time steps and affects the state extraction of each step. These two parameters should be selected carefully according to the specific applications to guarantee better results.

In the test scenarios of multiple objects, it can be concluded that although the robust birth policy may bring about larger cardinality error at the beginning of each track because of the postponed birth, it actually improves the object initialization and thus the following state estimations. After the missing states have been recovered by the backward filter, the final results show great superiority in performance compared to the naive LMB smoother.

## 5.2 Perspective

The proposed method in this thesis provides a viable solution to reference data generation for validating tracking applications. However, further effort is required to extend the method to achieve higher reliability and enhanced performance in tracking.

First of all, in the update step of the LMB filter, the local polar angle of a measurement point is simply computed based on the predicted position of the reference point, which may lead to incorrect correspondence between the true polar angle and the one used in the update. This usually happens when the object bypasses the sensor with only one visible side and changing velocity, and the velocity changes could sometimes not be properly captured for this reason. At least two solutions are currently conceivable to this problem: an extra step could be applied to align the measurements and the corresponding polar angles before performing update (e.g., considering the characteristics of the sensor, as in [75]); an alternative could be a coarse-to-fine update, i.e., the predicted state is first roughly shifted based on cluster registration (with e.g., [68, 76, 77] etc.) between the current and the previous steps, and the polar angles are assigned to the measurements according to the new reference point for a more precise update. These approaches could hopefully improve the performance of the proposed method especially for rapidly maneuvering objects.

Another extension could be made to the initial estimation of the reference point. In this thesis, the initial reference point is selected to be possibly near the geometric center of the object based on the point cluster in birthing. Prior knowledge of the contour shape could be incorporated for this task, and alternative methods could be investigated to further enhance the robustness of contour initialization.

Additionally, the replacement of the 2D star-convex contour model in this thesis with a 3D surface model would be of great interest, since most of the LiDAR-recordings are in the form of 3D point clouds and can provide more comprehensive information over the outer extent of an object. Possible 3D surface models are available in e.g., [78, 79] with similar inference processes as the 2D model described in this thesis.

Finally, it would be meaningful to apply the proposed method to more test scenarios from both simulation and real-world data collection to further investigate its parameterization, reliability, capability on different scenarios and for different types of objects, and processing efficiency etc., before it can be deployed into practical applications for reference data generation and system validation.



# List of Figures

1.1	Three processing layers of a typical automated vehicle (adapted from [2]).	1
1.2	SAE levels of automation for automated driving [3]. DDT denotes <i>Dynamic Driving Task</i> , OEDR denotes <i>Object and Event Detection and Response</i> , ODD denotes <i>Operational Design Domain</i> , ADS denotes <i>Automated Driving System</i> .	2
1.3	Basic idea of the RFS framework, including alteration in object amount, clutter and multiple measurement points from the same object [29]. . . . .	4
1.4	Example of track-before-detect states in a sequence of point cloud data, where solid and dashed lines denote tracked and untracked states. . . . .	4
2.1	The CTRV motion model . . . . .	9
2.2	The CTRA motion model . . . . .	11
2.3	Example of a star-convex shape and the corresponding radial function . . . . .	17
2.4	The exponential sine square kernel function with period $2\pi$ . . . . .	18
2.5	Example of contour approximation with Gaussian process at different levels of accuracy, controlled by the number of basis points $N^b$ and the length scale $l$ in Gaussian process regression. Basis points for all approximations are equidistantly spaced in range $(-\pi, \pi]$ . The first three approximations use length scale $l = 0.4$ , while the last one uses $l = 0.1$ . . . . .	19
3.1	The proposed method to generate track-before-detect estimations from a sequence of data $\{Z_k\}_{k=1}^T$ . $k_B$ denotes the original birth time of a certain track in the forward pass, and the possibly missing states at $k_B - 1, k_B - 2, \dots$ will be recovered by the backward filter. . . . .	23
3.2	Comparison of partition results from different clustering methods. As can be observed from the figures, k-means++ will mis-partition the measurement set when the two objects are close to each other, distance partitioning provides correct partitions but cannot exclude noise points that are close to the contour, which can be fixed by prediction pruning. . . . .	26
3.3	Examples of pair candidates for birthing. Different colors denote clusters from different time steps. . . . .	29
3.4	Two types of possible shapes formed by measurement points from LiDAR observation. . . . .	30
3.5	Factors influencing the suitability of a cluster regarding birthing: measurement density (calculated by the cluster cardinality and the azimuth angle range $\varphi$ ), mean distance $s$ to the sensor, and shape ratio (represented by the PCA variances $\sigma_1^2, \sigma_2^2$ ). . . . .	31

3.6	Examples of clusters receiving high existence probability from the quality measure, either due to high measurement density (the left one) or large shape ratio (the right one). . . . .	31
3.7	Example of hypothesis generation in the LMB-to-GLMB interpretation. The probability distributions of each hypothesis inherit from the involved tracks, and the corresponding hypothesis weight is calculated by multiplying the existence probabilities of all involved tracks. . . . .	33
3.8	Construction of a k-shortest path problem to generate the most significant hypotheses without exhaustive enumeration [20]. The presence of the node $\ell$ on the path from $S$ to $E$ indicates its presence in the corresponding hypothesis. . . . .	34
3.9	Example of the GLMB-to-LMB approximation. The probability distribution of each track in the LMB approximation is just the weighted sum of the distributions of the involved hypotheses in the GLMB, and the existence probability is calculated by summing up the corresponding weights. . . . .	35
3.10	Example of an object approaching the sensor from a distant position and the corresponding adjusting factors for existence probability regarding birthing. The track is initialized at time step 3 with two missing states. . . . .	39
4.1	Simulation results of tracking an extended object on a straight trajectory with the CV and CTRA motion models. . . . .	46
4.2	Simulation results of tracking a maneuvering extended object with the CTRV and CTRA motion models. . . . .	47
4.3	Test scenario of tracking a maneuvering extended object. . . . .	48
4.4	Simulation results of tracking a maneuvering extended object and the comparison between filtering and smoothing. . . . .	49
4.5	Influence of length scale selection in the kernel function on tracking a single maneuvering extended object. Solid lines are the final contour estimations, while the dashed lines are the reference contours. . . . .	50
4.6	True and estimated trajectories for tracking multiple extended objects (only the center of mass is shown for the position estimations). . . . .	51
4.7	OSPA error and IoU ratio of the naive smoother, S-RB and S-RB-BF. . . . .	53
4.8	Bird eye view of the testing ground and the trajectory of the target vehicle. . . . .	54
4.9	The test vehicle used for data acquisition and the LiDAR sensor setup on the roof structure. . . . .	55
4.10	True and estimated trajectories and contours, and the corresponding errors of the real-world experiment. . . . .	57
4.11	Number of points of the object cluster and the cardinality estimation with different $p_D$ in a certain interval. . . . .	58

# List of Tables

4.1	Average IoU ratios over time for each track . . . . .	52
4.2	Parameterization for the test on real-world data . . . . .	56



# Bibliography

- [1] D. Neuerer. *Selbstfahrende Autos - Dobrindt überzeugt Verbraucherschützer ein bisschen*. Jan. 2017. URL: <https://www.handelsblatt.com/politik/deutschland/selbstfahrende-autos-dobrindt-ueberzeugt-verbraucherschuetzer-ein-bisschen/19302674.html>. (accessed: 29.04.2020).
- [2] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus, and M. H. Ang. "Perception, planning, control, and coordination for autonomous vehicles." In: *Machines* 5.1 (2017), p. 6.
- [3] SAE J3016-2018. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Standard. United States: SAE International, June 2018.
- [4] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner. "Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions." In: *Proc. of the 18th IEEE International Conference on Intelligent Transportation Systems*. 2015, pp. 1455–1462.
- [5] Y. Takeda, C. Tsuchiya, and A. Khiat. "Ground truth generation for quantitative performance evaluation of localization methods in urban areas." In: *IEEE Intelligent Vehicles Symposium*. 2019, pp. 1152–1158.
- [6] L. Zhang, J. Lan, and X. R. Li. "Performance evaluation of multi-target tracking without knowing ground truth." In: *Proc. of the 19th IEEE International Conference on Information Fusion*. 2016, pp. 185–192.
- [7] C. Spampinato, S. Palazzo, and D. Giordano. "Evaluation of tracking algorithm performance without ground-truth data." In: *Proc. of the 19th IEEE International Conference on Image Processing*. 2012, pp. 1345–1348.
- [8] J. E. Stellet, L. Walkling, and J. M. Zöllner. "Post processing of laser scanner measurements for testing advanced driver assistance systems." In: *Proc. of the 19th IEEE International Conference on Information Fusion*. 2016, pp. 1999–2006.
- [9] U. Lages, M. Spencer, and R. Katz. "Automatic scenario generation based on laserscanner reference data and advanced offline processing." In: *IEEE Intelligent Vehicles Symposium*. 2013, pp. 153–155.
- [10] S. Blackman and R. Popoli. *Design and analysis of modern tracking systems*. Norwood, MA, USA: Artech House, 1999.
- [11] R. P. Mahler. *Statistical multisource-multitarget information fusion*. Norwood, MA, USA: Artech House, 2007.

## Bibliography

---

- [12] B. T. Vo. "Random finite sets in multi-object filtering." PhD thesis. University of Western Australia, June 2008.
- [13] T. Fortmann, Y. Bar-Shalom, and M. Scheffe. "Sonar tracking of multiple targets using joint probabilistic data association." In: *IEEE Journal of Oceanic Engineering* 8.3 (1983), pp. 173–184.
- [14] M. Mallick, S. Coraluppi, C. Carthel, V Krishnamurthy, and B Vo. "Multitarget tracking using multiple hypothesis tracking." In: *Integrated Tracking, Classification, and Sensor Management: Theory and Applications*. New York: Wiley Online Library, 2012, pp. 165–201.
- [15] S. S. Blackman. "Multiple hypothesis tracking for multiple target tracking." In: *IEEE Aerospace and Electronic Systems Magazine* 19.1 (2004), pp. 5–18.
- [16] R. P. Mahler. "Multitarget Bayes filtering via first-order multitarget moments." In: *IEEE Transactions on Aerospace and Electronic systems* 39.4 (2003), pp. 1152–1178.
- [17] B.-N. Vo and W.-K. Ma. "The Gaussian mixture probability hypothesis density filter." In: *IEEE Transactions on signal processing* 54.11 (2006), pp. 4091–4104.
- [18] B.-T. Vo, B.-N. Vo, and A. Cantoni. "The cardinality balanced multi-target multi-Bernoulli filter and its implementations." In: *IEEE Transactions on Signal Processing* 57.2 (2008), pp. 409–423.
- [19] B.-T. Vo and B.-N. Vo. "Labeled random finite sets and multi-object conjugate priors." In: *IEEE Transactions on Signal Processing* 61.13 (2013), pp. 3460–3475.
- [20] B.-N. Vo, B.-T. Vo, and D. Phung. "Labeled random finite sets and the Bayes multi-target tracking filter." In: *IEEE Transactions on Signal Processing* 62.24 (2014), pp. 6554–6567.
- [21] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer. "The labeled multi-Bernoulli filter." In: *IEEE Transactions on Signal Processing* 62.12 (2014), pp. 3246–3260.
- [22] D. Meissner, S. Reuter, and K. Dietmayer. "Road user tracking at intersections using a multiple-model PHD filter." In: *IEEE Intelligent Vehicles Symposium*. 2013, pp. 377–382.
- [23] R. Hoseinnezhad, B.-N. Vo, and B.-T. Vo. "Visual tracking in background subtracted image sequences via multi-Bernoulli filtering." In: *IEEE Transactions on Signal Processing* 61.2 (2012), pp. 392–397.
- [24] H. Deusch, S. Reuter, and K. Dietmayer. "The labeled multi-Bernoulli SLAM filter." In: *IEEE Signal Processing Letters* 22.10 (2015), pp. 1561–1565.
- [25] R. P. Mahler, B.-T. Vo, and B.-N. Vo. "Forward-backward probability hypothesis density smoothing." In: *IEEE Transactions on Aerospace and Electronic Systems* 48.1 (2012), pp. 707–728.
- [26] M. Beard, B. T. Vo, and B.-N. Vo. "Generalised labelled multi-Bernoulli forward-backward smoothing." In: *Proc. of the 19th International Conference on Information Fusion*. IEEE. 2016, pp. 688–694.

- [27] S. Wong, B. T. Vo, and F. Papi. "Bernoulli forward-backward smoothing for track-before-detect." In: *IEEE Signal Processing Letters* 21.6 (2014), pp. 727–731.
- [28] B.-T. Vo, D. Clark, B.-N. Vo, and B. Ristic. "Bernoulli forward-backward smoothing for joint target detection and tracking." In: *IEEE Transactions on Signal Processing* 59.9 (2011), pp. 4473–4477.
- [29] B.-n. Vo, M. Mallick, Y. Bar-shalom, S. Coraluppi, R. Osborne III, R. Mahler, and B.-t. Vo. "Multitarget tracking." In: *Wiley Encyclopedia of Electrical and Electronics Engineering* (1999), pp. 1–15.
- [30] M. Baum, V. Klumpp, and U. D. Hanebeck. "A novel Bayesian method for fitting a circle to noisy points." In: *Proc. of the 13th International Conference on Information Fusion*. IEEE. 2010, pp. 1–6.
- [31] N. Petrov, A. Gning, L. Mihaylova, and D. Angelova. "Box particle filtering for extended object tracking." In: *Proc. of the 15th International Conference on Information Fusion*. IEEE. 2012, pp. 82–89.
- [32] K. Granström, C. Lundquist, and U. Orguner. "Tracking rectangular and elliptical extended targets using laser measurements." In: *Proc. of the 14th International Conference on Information Fusion*. IEEE. 2011, pp. 1–8.
- [33] K. Granström, S. Reuter, D. Meissner, and A. Scheel. "A multiple model PHD approach to tracking of cars under an assumed rectangular shape." In: *Proc. of the 17th International Conference on Information Fusion*. IEEE. 2014, pp. 1–8.
- [34] J. W. Koch. "Bayesian approach to extended object and cluster tracking using random matrices." In: *IEEE Transactions on Aerospace and Electronic Systems* 44.3 (2008), pp. 1042–1059.
- [35] K. Granstrom and U. Orguner. "A PHD filter for tracking multiple extended targets using random matrices." In: *IEEE Transactions on Signal Processing* 60.11 (2012), pp. 5657–5671.
- [36] M. Beard, S. Reuter, K. Granström, B.-T. Vo, B.-N. Vo, and A. Scheel. "Multiple extended target tracking with labeled random finite sets." In: *IEEE Transactions on Signal Processing* 64.7 (2015), pp. 1638–1653.
- [37] M. Baum and U. D. Hanebeck. "Random hypersurface models for extended object tracking." In: *IEEE International Symposium on Signal Processing and Information Technology*. 2009, pp. 178–183.
- [38] M. Baum and U. D. Hanebeck. "Extended object tracking with random hypersurface models." In: *IEEE Transactions on Aerospace and Electronic Systems* 50.1 (2014), pp. 149–159.
- [39] M. Baum and U. D. Hanebeck. "Shape tracking of extended objects and group targets with star-convex RHMs." In: *Proc. of the 14th International Conference on Information Fusion*. IEEE. 2011, pp. 1–8.

## Bibliography

---

- [40] M. F. Huber. "Recursive Gaussian process: On-line regression and learning." In: *Pattern Recognition Letters* 45 (2014), pp. 85–91.
- [41] M. A. Osborne. "Bayesian Gaussian processes for sequential prediction, optimisation and quadrature." PhD thesis. Oxford University, 2010.
- [42] J. Hartikainen and S. Särkkä. "Kalman filtering and smoothing solutions to temporal Gaussian process regression models." In: *Proc. of the International Workshop on Machine Learning for Signal Processing*. IEEE. 2010, pp. 379–384.
- [43] N. Wahlström and E. Özkan. "Extended target tracking using Gaussian processes." In: *IEEE Transactions on Signal Processing* 63.16 (2015), pp. 4165–4178.
- [44] T. Hirscher, A. Scheel, S. Reuter, and K. Dietmayer. "Multiple extended object tracking using Gaussian processes." In: *Proc. of the 19th International Conference on Information Fusion*. IEEE. 2016, pp. 868–875.
- [45] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Norwood, MA, USA: Artech house, 2003.
- [46] D. Salmond and H Birch. "A particle filter for track-before-detect." In: *Proc. of the 2001 American Control Conference*. Vol. 5. IEEE. 2001, pp. 3755–3760.
- [47] Y. Boers, H. Driessens, J. Torstensson, M. Trieb, R. Karlsson, and F. Gustafsson. "Track-before-detect algorithm for tracking extended targets." In: *IEE Proceedings - Radar, Sonar and Navigation* 153.4 (2006), pp. 345–351.
- [48] S. J. Davey, M. G. Rutten, and B. Cheung. "A comparison of detection performance for several track-before-detect algorithms." In: *Proc. of the 11th International Conference on Information Fusion*. IEEE. 2008, pp. 1–8.
- [49] T. Ogawa and G. Wanielik. "Track-Before-Detect approach on LiDAR signal processing for low SNR target detection." In: *IEEE Intelligent Vehicles Symposium*. 2016, pp. 676–682.
- [50] X. R. Li and V. P. Jilkov. "Survey of maneuvering target tracking. Part I. Dynamic models." In: *IEEE Transactions on Aerospace and Electronic Systems* 39.4 (2003), pp. 1333–1364.
- [51] R. Schubert, E. Richter, and G. Wanielik. "Comparison and evaluation of advanced motion models for vehicle tracking." In: *Proc. of the 11th International Conference on Information Fusion*. IEEE. 2008, pp. 1–6.
- [52] B. D. Anderson and J. B. Moore. *Optimal filtering*. NJ, Englewood Cliffs: Prentice-Hall, 1979.
- [53] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. New York: Wiley, 2001.
- [54] F. Gustafsson and A. J. Isaksson. "Best choice of coordinate system for tracking coordinated turns." In: *Proc. of the 35th Conference on Decision and Control*. Vol. 3. IEEE. 1996, pp. 3145–3150.

- [55] F. Gustafsson. *Adaptive filtering and change detection*. New York: Wiley, 2000.
- [56] M. Tsogas, A. Polychronopoulos, and A. Amditis. "Unscented Kalman filter design for curvilinear motion models suitable for automotive safety applications." In: *Proc. of the 7th International Conference on Information Fusion*. IEEE. 2005, pp. 1295–1302.
- [57] R. P. Mahler. "'Statistics 101' for multisensor, multitarget data fusion." In: *IEEE Aerospace and Electronic Systems Magazine* 19.1 (2004), pp. 53–64.
- [58] R. P. Mahler. "'Statistics 102' for multisource-multitarget detection and tracking." In: *IEEE Journal of Selected Topics in Signal Processing* 7.3 (2013), pp. 376–389.
- [59] D. J. Daley and D. Vere-Jones. *An introduction to the theory of point processes*. New York: Springer Verlag, 1988.
- [60] S. Särkkä. *Bayesian filtering and smoothing*. Cambridge, UK: Cambridge University Press, 2013.
- [61] K. Granström, M. Baum, and S. Reuter. "Extended target tracking using Gaussian processes." In: *Journal of Advances in Information Fusion* 12.2 (2017), pp. 139–174.
- [62] C. E. Rasmussen and C. K. Williams. *Gaussian processes for machine learning*. MIT press, 2006.
- [63] D. J. MacKay. "Introduction to Gaussian processes." In: *Neural Networks and Machine Learning*. Ed. by C. M. Bishop. Springer Verlag, 1998, pp. 133–165.
- [64] C. M. Bishop. *Pattern recognition and machine learning*. New York: Springer, 2006.
- [65] D. Arthur and S. Vassilvitskii. "K-means++: The advantages of careful seeding." In: *Proc. of the ACM-SIAM Symposium on Discrete Algorithms*. Jan. 2007, pp. 1027–1035.
- [66] K. Granström, C. Lundquist, and U. Orguner. "A Gaussian mixture PHD filter for extended target tracking." In: *Proc. of the 13th International Conference on Information Fusion*. IEEE. Edinburgh, UK, July 2010.
- [67] K. Granstrom, C. Lundquist, and O. Orguner. "Extended target tracking using a Gaussian-mixture PHD filter." In: *IEEE Transactions on Aerospace and Electronic Systems* 48.4 (2012), pp. 3268–3286.
- [68] P. J. Besl and N. D. McKay. "A method for registration of 3-D shapes." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 239–256.
- [69] P. J. Huber. "Robust estimation of a location parameter." In: *Annals of Mathematical Statistics* 35.1 (Mar. 1964), pp. 73–101.
- [70] A. R. Runnalls. "Kullback-Leibler approach to Gaussian mixture reduction." In: *IEEE Transactions on Aerospace and Electronic Systems* 43.3 (2007), pp. 989–999.
- [71] H. E. Rauch, F Tung, and C. T. Striebel. "Maximum likelihood estimates of linear dynamic systems." In: *AIAA Journal* 3.8 (1965), pp. 1445–1450.

## Bibliography

---

- [72] D. Schuhmacher, B.-T. Vo, and B.-N. Vo. "A consistent metric for performance evaluation of multi-object filters." In: *IEEE Transactions on Signal Processing* 56.8 (2008), pp. 3447–3457.
- [73] S. Katz, A. Tal, and R. Basri. "Direct visibility of point sets." In: *ACM Transactions on Graphics (SIGGRAPH)* 26.3 (2007), 24–es.
- [74] Oxford Technical Solutions Ltd. *RT-Range user manual*. URL: <https://www.oxts.com/app/uploads/2019/12/RT-Range-manual-191203.pdf>. (accessed: 08.06.2020).
- [75] M. Michaelis, P. Berthold, T. Lüttel, D. Meissner, and H.-J. Wünsche. "Extended object tracking with an improved measurement-to-contour association." In: *Proc. of the 23th International Conference on Information Fusion*. accepted for publication. IEEE. Sun City, South Africa (Virtual Conference), June 2020.
- [76] P. Biber and W. Straßer. "The normal distributions transform: A new approach to laser scan matching." In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 3. 2003, pp. 2743–2748.
- [77] A. Myronenko and X. Song. "Point set registration: Coherent point drift." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.12 (2010), pp. 2262–2275.
- [78] F. Ebert and H.-J. Wünsche. "Dynamic object tracking and 3D surface estimation using Gaussian processes and extended Kalman filter." In: *Proc. of the 22th IEEE International Conference on Intelligent Transportation Systems*. IEEE. 2019, pp. 1122–1127.
- [79] A. Zea, F. Faion, and U. D. Hanebeck. "Tracking extended objects using extrusion Random Hypersurface Models." In: *Sensor Data Fusion: Trends, Solutions, Applications* (2014), pp. 1–6.