

VARIATIONAL BAYES METHODS FOR RANDOM MATRIX-BASED
EXTENDED TARGET TRACKING

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

BY

BARKIN TUNCER

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

FEBRUARY 2022

Approval of the thesis:

**VARIATIONAL BAYES METHODS FOR RANDOM MATRIX-BASED
EXTENDED TARGET TRACKING**

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

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

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

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

Examining Committee Members:

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

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

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

Prof. Dr. Orhan Arıkan
Electrical and Electronics Engineering Department, Bilkent Uni.

Assist. Prof. Dr. Sinan Yıldırım
Faculty of Engineering and Natural Sciences, Sabancı University

Date:

16.02.2021



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

Name, Last Name: Barkin Tuncer

Signature :

ABSTRACT

VARIATIONAL BAYES METHODS FOR RANDOM MATRIX-BASED EXTENDED TARGET TRACKING

Tuncer, Barkın

M.S., Department of Electrical and Electronics Engineering

Supervisor : Assoc. Prof. Dr. Emre Özkan

February 2022, 130 pages

In this thesis, we investigate two major topics; Random matrix-based models using variational Bayes inference for extended target tracking and shape classification algorithms for tracking applications. In the scope of this thesis, we have derived two novel random matrix-based tracking algorithms. First, to represent the extent of dynamic objects as an ellipsoid with a time-varying orientation angle, and secondly, to estimate the extent of the object or a group of objects with more than one ellipsoid. In both of these solutions, we have used the variational Bayes technique to perform approximate inference, where the Kullback-Leibler divergence between the true and the approximate posterior is minimized by performing fixed-point iterations. The update equations are easy to implement, and the algorithms can be used in real-time tracking applications. We illustrated the performance of the methods in simulations and experiments with real data. In extended target tracking, once the shape estimate of an object is formed, it can naturally be utilized by high-level tasks such as classification of the object type. Therefore, we present a naively deep neural network, which

consists of one input, two hidden and one output layers, to classify dynamic objects regarding their shape estimates. In this manner, the proposed method shows superior performance in comparison to a Bayesian classifier for simulation experiments.

Keywords: Extended Target Tracking, Object Tracking, Variational Bayes, Kalman Filter, Machine Learning, Classification, Target Classification, Neural Networks



ÖZ

RASLANTISAL MATRİS TABANLI GENİŞLETİLMİŞ HEDEF TAKİP MODELLERİ İÇİN VARYASYONEL BAYES YÖNTEMLERİ

Tuncer, Barkın

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Doç. Dr. Emre Özkan

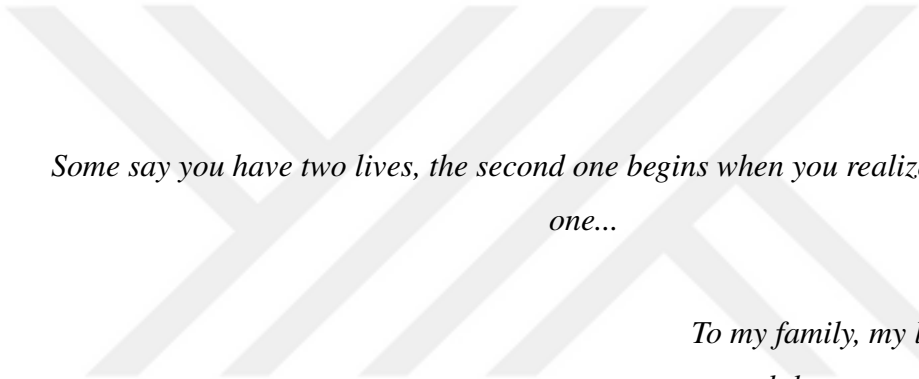
Şubat 2022, 130 sayfa

Bu tezde iki ana konu incelenmiştir; Genişletilmiş hedef takibi için varyasyonel Bayes çıkarımı kullanan rastlantısal matris tabanlı modeller ve hedef takibi uygulamaları için obje sınıflandırma algoritmaları. Bu tez kapsamında iki yeni rastlantısal matris tabanlı hedef takibi algoritması türetilmiştir. Birincisi, dinamik nesnelerin kapsamını zamanla değişen bir oryantasyon açısına sahip bir elipsoid olarak temsil eden bir algoritmadır. İkinci yaklaşımda ise, bir nesnenin veya bir grup nesnenin kapsamı birden fazla elipsoid ile temsil edilmektedir. Bu çözümlerin her ikisinde de, yaklaşık çıkarımı gerçekleştirmek için varyasyonel Bayes tekniği kullanılmıştır; burada gerçek ve yaklaşık sonsal olasılık arasındaki Kullback-Leibler ayrışmasının, sabit nokta yinelemeleriyle en aza indirilmesi amaçlanmıştır. Güncelleme denklemlerinin uygulanması kolaydır ve algoritmalar gerçek zamanlı izleme uygulamalarında kullanılabilir haldedir. Simülasyon ve deneylerde gerçek veriler kullanılarak yöntemlerin performansı gösterilmiştir. Genişletilmiş hedef izlemede, bir nesnenin şekil tahmini bir kez oluş-

turulduktan sonra, nesne türünün sınıflandırılması gibi üst düzey görevler için kullanılabilir. Bu nedenle, dinamik nesneleri şekil tahminlerine göre sınıflandırmak için bir girdi, iki gizli ve bir çıktı katmanından oluşan saf derin bir yapay sinir ağı algoritması sunulmuştur. Bu şekilde önerilen yöntemin, simülasyon deneylerinde başka bir Bayes sınıflandırıcısına kıyasla daha üstün performans gösterdiği gözlemlenmiştir.

Anahtar Kelimeler: Genişletilmiş Hedef Takibi, Obje Takibi, Varyasyonel Bayes, Kalman Filtresi, Makine Öğrenmesi, Sınıflandırma, Hedef Sınıflandırması, Yapay Sinir Ağları





*Some say you have two lives, the second one begins when you realize you have only
one...*

*To my family, my love, my friends,
and the memory of my father...*

ACKNOWLEDGMENTS

To begin with, I would like to thank myself for believing in me, accomplishing all the hard work, having no days off, and never giving up.

I would love to express my deepest gratitude and my humblest regards to my esteemed supervisor Assoc. Prof. Dr. Emre Özkan for his never-ending support, for continuously broadening my horizons, and for providing me this excellent opportunity to involve in academia.

I cannot express enough how much I appreciate Dr. Murat Kumru's support, both as an academic mentor and as a friend, not only when I stuck at some point during my research, but also in making decisions about my life.

I am also sincerely grateful to the METU Sensor Fusion group for all of their efforts in my research endeavors and throughout my graduate education.

I bestow my thankfulness to my friends for always cheering me up, especially the group with whom I had a lot of amazing and joyful adventures, Cina-i Şebeke.

I would like to verbalize my enormous thanks to my family, Arzu Ateş, Yalım Ateş, Barış Tuncer, Emre Ateş, Deren Ateş, for always supporting me no matter what and bearing with me during these difficult years.

Last but foremost, I owe my love, Ceyda Pınar, a great debt of gratitude for always listening to me, supporting me with my decisions, and being there for me in my stressful times.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS	xi
LIST OF TABLES	xvii
LIST OF FIGURES	xix
LIST OF ALGORITHMS	xxiii
LIST OF ABBREVIATIONS	xxiv

CHAPTERS

1	INTRODUCTION	1
1.1	Motivation	1
1.2	Extended Target Tracking	2
1.3	Classification of Targets	5

1.4	Contribution	6
1.5	Thesis Outline	7
2	BACKGROUND	9
2.1	State Estimation	9
2.2	Recursive Bayesian State Estimation	10
2.2.1	Prediction Update	10
2.2.2	Measurement Update	11
2.3	Kalman Filtering	12
2.4	Bayesian Conjugacy	13
2.5	Variational Bayes Approach	14
2.5.1	Factorized Distributions	15
2.6	Random Matrix Model	17
2.7	Machine learning with Neural Networks	19
2.7.1	Learning Types in Machine Learning	19
2.7.1.1	Supervised Learning	21
2.7.2	Biological Interpretation of Neural Networks	21
2.7.3	Neural Network Architecture	21
2.7.3.1	Input Layer	22

2.7.3.2	Output Layer	22
2.7.3.3	Hidden Layer	23
2.7.3.4	Learning Mechanism	23
2.8	Publications	27
3	RANDOM MATRIX BASED EXTENDED TARGET TRACKING WITH ORIENTATION: A NEW MODEL AND INFERENCE	29
3.1	Introduction	29
3.2	Problem Definition	34
3.3	Measurement Update	37
3.3.1	Variational Inference	38
3.3.2	Calculation of $q_{\mathbf{x}}^{(\ell+1)}(\cdot)$	39
3.3.3	Calculation of $q_{\mathbf{x}}^{(\ell+1)}(\cdot)$	40
3.3.4	Calculation of $q_{\mathbf{z}}^{(\ell+1)}(\cdot)$	41
3.3.5	Calculation of $q_{\theta}^{(\ell+1)}(\cdot)$	42
3.3.6	Expectation Calculations	44
3.3.7	Calculation of $\mathbb{E}_{q_{\mathbf{x}}^{(\ell)}, q_{\theta}^{(\ell)}} [(\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T)^{-1}]$	45
3.4	Time Update	47
3.5	A Closer Look to a Single Measurement Update	48

3.6	Experimental Results	50
3.6.1	Simulations	50
3.6.2	Constant Velocity Model	51
3.6.3	Experimental Trajectory	54
3.6.4	Real Data Experiment	57
3.7	Conclusion and Discussion	59
4	MULTI-ELLIPSOIDAL EXTENDED TARGET TRACKING WITH VARIATIONAL BAYES INFERENCE	61
4.1	Introduction	61
4.2	Problem Formulation	64
4.3	Measurement Update	68
4.3.1	Calculation of $q_{\mathbf{z}}^{(i+1)}(\cdot)$	72
4.3.2	Calculation of $q_{\mathbf{r}}^{(i+1)}(\cdot)$	73
4.3.3	Calculation of $q_{\mathbf{x}}^{(i+1)}(\cdot)$	73
4.3.4	Calculation of $q_{\mathbf{x}}^{(i+1)}(\cdot)$	74
4.3.5	Calculation of $q_{\pi}^{(i+1)}(\cdot)$	74
4.4	Time Update	76
4.5	Experimental Results	77

4.5.1	Simulations	79
4.5.2	Occlusion Scenario	84
4.5.3	Experiments with Real Data	85
4.6	Discussions	88
4.6.1	The number of sub-objects	88
4.7	Conclusion and Future Work	89
5	EXTENDED TARGET TRACKING AND CLASSIFICATION USING NEURAL NETWORKS	91
5.1	Introduction	91
5.2	Object Classification	93
5.2.1	Feature Selection for Contour Representation	93
5.2.2	Classifier Architecture	96
5.3	GP-ETT	98
5.4	Simulations and Results	101
5.5	Conclusion and Future Work	106
6	CONCLUSION AND FUTURE WORK	107
	REFERENCES	109

.1	Appendix	119
.1.1	Derivation of Variational Updates	119
.1.1.1	Calculation of $q_{\mathbf{z}}^{(i+1)}(\cdot)$	119
.1.1.2	Calculation of $q_{\mathbf{r}}^{(i+1)}(\cdot)$	120
.1.1.3	Calculation of $q_{\mathbf{x}}^{(i+1)}(\cdot)$	122
.1.1.4	Calculation of $q_{\mathbf{X}}^{(i+1)}(\cdot)$	124
.1.1.5	Calculation of $q_{\pi}^{(i+1)}(\cdot)$	125
.1.2	Expected Values Required for Iterations	126
.1.3	Calculation of δ and Δ	127
.1.4	Proof of Lemma 1	128
.1.5	The parameters of the experiment in section 3.5	129
.1.6	An example for the collapsing extent estimates	130

LIST OF TABLES

TABLES

Table 3.1	Notations	33
Table 3.1	Notations (continued)	34
Table 3.2	The GW distance values for Gaussian measurements.	52
Table 3.3	The GW distance values for uniformly distributed measurements. . .	52
Table 3.4	The heading angle RMSE values for Gaussian and uniform mea- surements.	53
Table 3.5	The GW distance and heading angle RMSE values of the scenario in Section 3.6.3 when the measurements are uniformly distributed. . . .	55
Table 3.6	The GW distance and heading angle RMSE values of the scenario in Section 3.6.3 when the measurements are generated from a Gaussian distribution.	55
Table 4.1	Notations	65
Table 4.1	Notations (continued)	66
Table 4.2	The simulation parameters for the first scenario.	81
Table 4.3	The RMSE, IOU, GW values and the computation times for the first scenario.	82
Table 4.4	The simulation parameters for the second scenario.	83

Table 4.5 The RMSE, IOU, GW values and the computation times for the second scenario.	84
Table 5.1 Classification results on the synthetic data set in terms of accuracy rate and the execution times for each shape.	104
Table 5.2 Some of the results that NN shows superior performance against Bayesian approach.	105



LIST OF FIGURES

FIGURES

Figure 2.1 Inverse Wishart distribution sampling and measurements visualization examples	18
Figure 2.2 An illustration of the RM based method proposed in [78].	19
Figure 2.3 A basic biological representation of a neuron.	22
Figure 2.4 An example neural network model.	24
Figure 2.5 A deeper look of a single neuron in a hidden layer	24
Figure 2.6 An illustration of the backpropagation algorithm.	26
Figure 3.1 An illustration of the target extent model with ellipsoids corresponding to the covariance matrices $\mathbf{X}_k$ (dashed line), $\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T$ (solid line) and $s \mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T + \mathbf{R}$ (dotted line), respectively.	36
Figure 3.2 A single measurement update for VB and the EKF approach. The prior and posterior mean shape estimates are represented by green dotted and solid lines for VB, respectively. The red dashed line indicates the prior mean shape estimate while the red solid line depicts the posterior mean estimate for EKF approach. The VB # i denotes the i^{th} variational iteration shape estimate mean of the VB algorithm.	49
Figure 3.3 An example MC run of the scenario in Section 3.6.3	53

Figure 3.4	The outline of the movement of the vehicle during the time-lapse. The vehicle starts from the dark blue colored parking spot; and follows the colored path until the red colored parking spot. In the figure, the last frame is shown.	56
Figure 3.5	A representative MC run of the real data experiment. The extent estimates of VB, Algorithm-1, Algorithm-2, and GP-ETT are shown in black, red, blue, and green lines, respectively. The measurements are represented with green dots.	56
Figure 4.1	An extended target representation with multiple ellipses (solid lines). Classical RM models represent the same target extent as a single ellipse (dashed black line). The measurements are shown with red stars.	62
Figure 4.2	Illustration of the extent estimation for $L = 4$ where + sign denotes the measurements.	64
Figure 4.3	The estimation results for the first scenario where two co-centered objects move along a straight line. The extent estimates of VB, VPF, and LL are shown in purple, orange, and yellow lines, respectively. The ground truth is depicted by black dashed lines.	80
Figure 4.4	The estimation results for the second scenario where four sub-objects with distinct extents move along a straight line. The extent estimates of VB, VPF, and LL are shown in purple, orange, and yellow lines, respectively. The ground truth is depicted by black dashed lines.	80
Figure 4.5	The extent estimates of VB (purple solid line) and FFK (yellow solid line) together with the ground truth (black dashed line). The top two targets are occluded between $t = 36$ and $t = 51$. During this interval, no measurements are acquired from these targets. The occlusion region is represented by the green area.	82

Figure 4.6	The results of experiments with real data: (a-b) Delta wing airplane (c-d) Glider. In the last frame of each scenario, the result of the single ellipsoidal target tracking algorithms [90] (magenta dashed line) and [18] (yellow dashed line) are also presented.	87
Figure 4.7	An illustration of the results of the variational Gaussian mixture estimation algorithm given in [8, Section 10.2] for selecting the number of sub-objects of the plane-shaped object.	88
Figure 4.8	An illustration of the results of the variational Gaussian mixture estimation algorithm given in [8, Section 10.2] for selecting the number of sub-objects in the V-shape formation.	88
Figure 4.9	An illustration of the results of the variational Gaussian mixture estimation algorithm given in [8, Section 10.2] for selecting the number of sub-objects at a given time during the motion.	89
Figure 5.1	Block diagram of the proposed online classification algorithm. . . .	92
Figure 5.2	Visualization of the parameters used in the feature extraction process. The solid red curve represents the contour of the object. The minimum bounding circle and rectangle are plotted in green and yellow, respectively. Major and minor axes of the minimum bounding rectangle are denoted by l_1 and l_2 . The blue dashed line indicates the convex hull of the object.	95
Figure 5.3	A simple neural network including one input layer, one hidden layer and one output layer.	97
Figure 5.4	An example star-convex contour described by radial function $r = f(\theta)$, [82].	99
Figure 5.5	The covariance function of the GP as defined in (5.6).	99

Figure 5.6 Typical contour estimates produced by GP-ETT. (Contour estimates are plotted in blue while red crosses represent point measurements.) 102

Figure 5.7 The performance curves of the proposed model (NN-feature). The network parameters are set to be the values satisfying the best performance in the validation set. 103

Figure .1 The visualization of the collapsing behavior of Algorithm-1. Blue stars represent the measurements, the solid green and red lines stand for the posterior means of the VB and EKF updates, respectively. The solid black line indicates the median of the true posterior, which is computed by using 1 million Monte Carlo samples. 130



LIST OF ALGORITHMS

ALGORITHMS

Algorithm 3.1	Variational Measurement Update	46
Algorithm 4.1	Measurement Update for ME-ETT Model	74



LIST OF ABBREVIATIONS

ABBREVIATIONS

GP	Gaussian Process
KF	Kalman Filter
EKF	Extended Kalman Filter
PSDM	Positive Semi-Definite Matrix
RM	Random Matrix
RMM	Random Matrix Model
PDF	Probability Density Function
RMSE	Root Mean Square Error
IOU	Intersection over Union
ETT	Extended Target Tracking
MC	Monte Carlo
GW	Gaussian Wassertein
$\mathcal{N}$	Normal Distribution

CHAPTER 1

INTRODUCTION

1.1 Motivation

Is it possible to build an autonomous vehicle? Will we ever be able to experience the joy of a vehicle driving itself from a starting point to a destination without any interference from the driver? Today, people ask questions similar to these every day and are curious about the answers. In fact, even if we seem to be approaching that standard, there is still a long way to go. According to the report of the Society of Automotive Engineers (SAE International), there are six levels of autonomous driving. The labeling of the layers starts with Level 0: No Automation. At this level, the driving experience heavily relies on the human control of the vehicle in every aspect. In Level 1: Driver Assistance level, the system starts to achieve simple tasks such as steering or acceleration/deceleration based on the knowledge of the driving environment. However, the driver still needs to accomplish all remaining tasks of driving. In Level 2: Partial Automation, as in Level 1, the system is able to steer along with acceleration and deceleration while the human monitors the behavior. Up to Level 2, the driver is required to monitor the driving environment; however, with Level 3: Conditional Automation, we enter the autonomous driving scheme. At Level 3, the driver is not supposed to monitor the environment yet; it is still a necessity to take control with a notice from the system. In Level 4: High Automation, the system monitors and drives in specific environments without having any inputs from the driver. If human interference is necessary for some situations and the driver does not respond appropriately, the system still performs well. From this level, the vehicles are started

to be called fully autonomous. At the last level, Level 5: Full Automation, the system can perform the driving action in all conditions without any attention from the driver.

Car manufacturing companies have been trying to exceed these levels one by one for years. As the current state of the vehicle rises, relative to the above-mentioned levels, the system becomes more complex, and the demand for better technology increases. To accomplish this goal, numerous independent subsystems have to be installed and set to work together. In this thesis, mainly, we focused on the perception branch, where we try to observe an object and its behavior by referring the information provided by the sensor measurements. In this context, we put emphasis on two specific tasks that are critical for perception; extended target tracking and target classification. The first subsystem, extended target tracking, is required in order to track the state of the objects around the environment. So that the vehicle can position and orient itself according to the movement of the surrounding objects. The second, classification of the objects, is needed to distinguish the objects in the environment from each other. Both of these systems are required to have a safe driving experience.

1.2 Extended Target Tracking

In essence, target tracking is the estimation of the state of a moving object based on remote observations of the object [2, Ch. 12]. The term *target* can be considered as an object whose state is interest of us, while the term *state* may consist of a collection of dynamic variables such as position, velocity, orientation, and angular velocity, which fully describe the system [70, Ch. 1].

In order to have reliable state estimations, we need to gather information from the target, i.e., we have to measure some quantities of the target. These variables, which rely on the type of the sensor, might include positional information, or velocity, etc. The data collection process is achieved via different kinds of sensors such as Light Detection and Ranging (LiDAR) sensor, radar, and camera.

Some difficulties must be overcome to track targets accurately. First of all, in most

of target tracking scenarios, the number of the targets is unknown. Even though this is the case for many situations, such as autonomous driving, applications can also be encountered where the target number is assigned to a particular value. Tracking of sports athletes in competitions may be given as examples of such problems [84, 85]. The second difficulty is that it is not known whether the source of the measurements gathered by the sensors is the target to be tracked. Since there is no information about the location of the target on the sensor side, the sensor collects every data it is able to. This behavior results in collecting clutter measurements which are measurements that the target does not cause [22].

In such a case, even if we know which of the measurements we collect originates from the target of interest, this time, the noise caused by the sensor or due to the environment affects the measurement we receive in a negative way. Because of this so-called measurement noise, a measurement could be identified as a clutter even if it comes from the target and vice versa.

Targets can be categorized according to two attributes; first, the dimensions according to the sensor resolution, and secondly, their distance from the sensor. Based on these characteristics, we can make the following two definitions:

- Point target: A target that can give rise at most a single noisy measurement in a single sensor scan [22].
- Extended target: A target that can potentially generate more than one mostly noisy measurement in a single measurement scan [22].

In modern target tracking, the resolution of the sensors and the application fields, such as tracking of vehicles, bicycles, etc., allow us to make extended target assumptions. Therefore, we will mainly focus on this aspect of the target tracking field. Although extended target tracking (ETT) introduces a new challenge, the uncertainty of the number of the measurements the target generates, it allows us to deduct some beneficial properties of the target, such as the size, shape, and orientation. These estimated characteristics of the target may positively affect tracking performance and may be used in further purposes such as classification.

An early approach to ETT consists of representing the target’s extent, sometimes called the contour, as a simple shape and estimating the relevant parameters. These simple shapes can be a line [23], a circle [6] or a rectangle [24]. In order to approximate the extent with a more complex shape, *random hyper-surface models* [4, 5, 90] could be used, which assume that the measurements are generated from an unknown random surface. A more recent proposal to the ETT problem, especially well-suited to the star convex shapes, is to use Gaussian Process (GP) models initiated by [82]. GPs rapidly managed to attract the attention of the target tracking community, and extensive research began to be carried out [1, 33, 43, 53]. GPs are also significantly considered in machine learning and deep learning applications [68] since it is straightforward to track the analytical properties and posterior computations. In GP-based ETT solutions, the target’s contour is described as an unknown radial function with a GP prior. Hence, the estimated contour can be denoted as a descriptive representation of the target, i.e., the radial function that is used to describe the target’s contour can be utilized to classify the targets [76, 77].

Another common approach in the ETT literature is the random matrix model (RMM) based algorithms [18, 25, 39, 42, 60, 78, 88–90], which was initiated by Koch [42]. In the RMM based methods, the extent of the target is described by an ellipse that is represented by an unknown semi-positive definite (SPD) matrix. The first study [42] describes an exact inference in the cost of neglecting the measurement noise. This model is later improved by [18] to regard the measurement noise in the measurement update at the expense of exact inference. Even if the resulting update equations in [18] are intuitive, the approximations needed are challenging to quantify theoretically. In [60], the variational Bayes technique is utilized in order to calculate approximate posteriors. To account for maneuvering of the target, the earlier works on RMM [18, 42, 60] rely on a forgetting factor which is supposed to increase the uncertainty of the extent approximation. However, more recent works [25, 78, 88–90] incorporate the orientation of the target as a separate random variable to estimate.

As we mentioned before, ETT algorithms naturally provide us with information about the shape of the target. This information might be useful in further research topics,

such as classification of targets. In the following section, we will give an introduction to how the classification problem is handled in conjunction with ETT frameworks.

1.3 Classification of Targets

One of the many reasons employing target tracking algorithms in various areas, such as autonomous driving, military, etc., is to use the output data in further deductions. Primitive approaches to the classification of tracked targets revolve around discriminating the dynamic behavior, motion cues, attributes, or fingerprints of the targets [3, 9, 20]. In one of the early works, authors of [13] proposed a *joint target tracking and classification* (JTC) framework, where the joint target state-class posterior density is ought to be estimated together. In [73], authors published an approach to the JTC problem based on belief functions. A batch iterative optimization algorithm that ought to minimize the Bayes risk consists of estimation and classification errors is presented in [47], and a recursive version of this method is introduced in [49]. A recent publication [11] also governs Bayes risk to solve the extended object classification problem. In most of these methods, a multi-model filtering technique is considered, and the classification output has a direct effect on the tracking performance of the algorithm, i.e., cross-coupled feedback is specified between the state and class.

On the other hand, with the recent developments in sensor systems, it is possible to acquire reliable contour estimates from ETT algorithms. Apart from the above-mentioned approaches, we can classify tracked targets in an online manner based on the contours predicted by such ETT methods [76, 77] without having a direct effect on the tracking performance. The authors in [77] solve this problem by extracting some descriptive features from the contour estimates and feeding them to a Bayesian classifier which also incorporates the uncertainty information hidden inside the covariance matrix of the contour estimate. In [76], a neural network is used to solve the classification problem considering the features gathered from the contour estimates.

1.4 Contribution

Contributions of this thesis can be listed as follows:

- First contribution:** Developed a novel random matrix model-based extended target tracking model which also estimates the heading-angle
- Second contribution:** Proposed a novel multi-ellipsoidal extended target tracking algorithm
- Third contribution:** Established a novel deep learning-based object classifier on top of a Gaussian process-based extended target tracker

In our first contribution, we proposed a novel random matrix model that describes the extent with inverse Gamma prior distributions and denotes the heading-angle with a Gaussian random variable. Defining the extent parameters with inverse Gamma priors ensures the positive semi-definiteness, which is crucial in the context of random matrix models. Since it is not possible to find a closed-form analytical expression for the posterior, we have utilized the variational Bayes inference to approximate the resulting posterior. We illustrated the performance of the method in simulations and experiments with real-life data. The proposed method outperforms the state-of-the-art methods when compared with respect to accuracy and robustness.

The second contribution includes a novel algorithm to represent a target's extent or a group of targets with multiple ellipses with respect to the random matrix model. In this work, each ellipse is represented by an unknown symmetric positive-definite random matrix. At its core, the proposed algorithm solves two challenging problems. The first one is the data association problem caused by the numbers of ellipses and multiple measurements. The second problem involves the non-conjugate priors and likelihoods, which requires special algorithms to approximate the resulting posterior distribution. We have utilized the variational Bayes method to solve the association problem and to approximate the intractable true posterior. The demonstrations of the

performance of the approach are done with simulations and real-life data experiments. The results of the experiments show that the proposed method significantly improves the computation time while achieving better kinematic and extent state estimation performance compared to the existing approaches in the literature.

Our third contribution focuses on a different aspect of perception. In this work, we leveraged the information that is the output of the extended target tracking algorithms and used that data to classify the targets online. We have used a Gaussian process-based extended target tracking algorithm since the extent estimate is descriptive. Using the output of the tracker, we have extracted and analyzed meaningful features from the extent estimate and supplied them into a shallow deep neural network model. The model consists of an input layer, 2 hidden layers, and a single output layer. Simulations show that the proposed method outperforms the previously published Bayesian classifier.

1.5 Thesis Outline

In this thesis, first of all, we will give a brief introduction to state estimation and Bayesian state estimation topics in 2.1, and 2.2, respectively. Then, we will extend this knowledge to Kalman filtering in 2.3. An essential aspect of the Kalman filters, which is Bayesian conjugacy, will be discussed in 2.4. Later, we will explain the details of a solution, namely the variational Bayes approach, to the conjugacy problem in Kalman filters in 2.5. Random matrix models, which is one of the most investigated branches of the extended Kalman filters, are introduced in 2.6. A compressed explanation of the neural networks is given in 2.7 as the last item for the Background Knowledge Chapter.

In Chapters 3-5, we will go over the publications which are published or submitted during the course of this thesis.



CHAPTER 2

BACKGROUND

2.1 State Estimation

Fundamentally, state estimation is a technique to find unknown values of so-called state variables by interpreting measurements affected by noise. Consider the state-space model below.

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \quad (2.1a)$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{v}_k) \quad (2.1b)$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the state vector that contains the values of state variables at time k with the initial state $\mathbf{x}_0 \sim p(\mathbf{x}_0)$. The measurement vector at time k is represented by $\mathbf{y}_k \in \mathbb{R}^{n_y}$. The functions $f(\cdot)$ and $h(\cdot)$ are the state transition function and the measurement function, respectively. In a perfect world, we would be able to estimate the state variables directly from the measurements if we had knowledge of the true process model. Unfortunately, in some cases, we cannot have precise information about the actual process model, which introduces an additional challenge to the problem. In addition to that, most measurements are affected by noise from the environment or the sensor itself. The noise component in (2.1a), $\mathbf{w}_{k-1}$, also known as white process noise, presents our lack of knowledge about the system model. If process noise level is considerably high, we cannot trust our state transition estimation. On the other hand, $\mathbf{v}_k$ denotes the white measurement noise. As for process noise, if measurement noise is significant, we cannot put our trust in measurements. In target tracking problems that we will cover in this thesis, we assume that the distribution of $\mathbf{w}_k \sim p(\mathbf{w}_k)$

and $\mathbf{v}_k \sim p(\mathbf{v}_k)$ are known, and they are independent of the state $\mathbf{x}_k$. Based on this information, our aim is to find the posterior density of the state $p(\mathbf{x}_k|\mathcal{Y}_{1:k})$ based on the measurements collected up to time k , $\mathcal{Y}_{1:k} \triangleq \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k\}$.

2.2 Recursive Bayesian State Estimation

In Bayesian state estimation, the posterior distribution is inferred by the prior and likelihood distributions using the Bayes rule (2.2), where $\mathbf{x}$ and θ are state vector and the parameters, respectively. In most of the target tracking applications, however, the model is an online dynamic system. For this reason, it is desirable to handle the problem with a recursive solution, in which new information is collected in a timely manner and to take into account the past information. To achieve this, with an initial state vector distribution $p(\mathbf{x}_0)$, we follow the scheme in (2.3).

$$p(\mathbf{x}|\theta) \propto p(\mathbf{x}) \times p(\theta|\mathbf{x}), \quad (2.2a)$$

$$posterior \propto prior \times likelihood. \quad (2.2b)$$

$$\dots \xrightarrow{\text{correction}} p(\mathbf{x}_{k-1}|\mathcal{Y}_{1:k-1}) \xrightarrow{\text{prediction}} p(\mathbf{x}_k|\mathcal{Y}_{1:k-1}) \xrightarrow{\text{correction}} p(\mathbf{x}_k|\mathcal{Y}_{1:k}) \xrightarrow{\text{prediction}} \dots \quad (2.3)$$

where $p(\mathbf{x}_k|\mathcal{Y}_{1:k-1})$ and $p(\mathbf{x}_k|\mathcal{Y}_{1:k})$ denote the predicted state density and posterior state density, respectively. Note that the one-step Markov property is assumed here, i.e., the current state depends only on the state of the previous time step. Between each successful cycle, we collect new measurements, $\mathcal{Y}_{k-1:k}$ from the sensor.

2.2.1 Prediction Update

In the prediction update, our goal is to compute the sufficient statistics of the predictive density by propagating the posterior density of the previous time step in time (2.4).

$$p(\mathbf{x}_{k-1}|\mathcal{Y}_{1:k-1}) \xrightarrow{\text{prediction}} p(\mathbf{x}_k|\mathcal{Y}_{1:k-1}) \quad (2.4)$$

This is calculated with the well-known Chapman-Kolmogorov equation (2.5), which essentially governs the law of total probability and Markov property to relate joint probability density functions (PDFs) [64].

$$p(\mathbf{x}_k|\mathcal{Y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathcal{Y}_{1:k-1})d\mathbf{x}_{k-1}, \quad (2.5)$$

In general, it is not straightforward to solve this equation and obtain a compact analytical form. However, under some assumptions, it is possible to obtain expressions regarding the propagation of sufficient statistics of the predictive distribution.

2.2.2 Measurement Update

If any measurements are obtained, we can utilize these measurements to gain further information about the states and combine it with the predicted density. This process is called the measurement update or correction.

$$p(\mathbf{x}_k|\mathcal{Y}_{1:k-1}) \xrightarrow{\text{correction}} p(\mathbf{x}_k|\mathcal{Y}_{1:k}) \quad (2.6)$$

Given all measurements up to time k , we can calculate the posterior density of the state variables by the well-known Bayes rule.

$$p(\mathbf{x}_k|\mathcal{Y}_{1:k}) = \frac{p(\mathbf{x}_k|\mathcal{Y}_{1:k-1})p(\mathbf{y}_k|\mathbf{x}_k)}{p(\mathbf{y}_k|\mathcal{Y}_{1:k-1})}, \quad (2.7)$$

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

where $p(\mathbf{x}_k|\mathcal{Y}_{1:k-1})$ is the predicted state density acquired from the prediction update, and $p(\mathbf{y}_k|\mathbf{x}_k)$ denotes the measurement likelihood density. The term in the denominator, $p(\mathbf{y}_k|\mathcal{Y}_{1:k-1})$, is independent of the state vector $\mathbf{x}_k$ and designated as the predictive likelihood distribution.

As in the prediction update, it might become a challenge to calculate the posterior distribution based on the PDFs. To compute the prediction and measurement update, there are several compact analytical formulas under certain assumptions. Some of the most well-known and used assumptions are gathered under the name Kalman filter, which will be explained next.

2.3 Kalman Filtering

The Kalman filter (KF) has been widely used in both industry and academia since its first publication in the 1960s [37] to the present day. In essence, KF is the optimal solution to the filtering problem given in the system (2.1) under linear and Gaussian assumptions. In other words, the state transition function $f(\cdot)$ and the measurement function $h(\cdot)$ are assumed to be linear in the state vector, and the noise terms are assumed to have zero mean Gaussian distribution. Therefore, the state space system in (2.1) simplifies to the system below.

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (2.9a)$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (2.9b)$$

where $\mathbf{F} \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{H} \in \mathbb{R}^{n_y \times n_x}$ are the state transition and measurement matrices, respectively. The process noise is a white Gaussian noise $\mathbf{w}_{k-1} \sim \mathcal{N}(0, \mathbf{Q})$. The measurement noise has also a white Gaussian distribution $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R})$.

In this manner, the prior distribution of the state vector at time $k = 0$ is assumed to follow a Gaussian distribution $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{m}_0, \mathbf{P}_0)$. In the prediction update, the sufficient statistics of the random state variables are propagated in time concerning the equation in (2.9a). The resulting predictive distribution is then updated according to the measurement equation in (2.9b).

Assume that we have the following prior and likelihood densities, respectively.

$$p(\mathbf{x}_{k-1|k-1} | \mathcal{Y}_{0:k-1}) = \mathcal{N}(\mathbf{x}_{k-1|k-1}; \mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \quad (2.10a)$$

$$p(\mathbf{x}_{k|k-1} | \mathcal{Y}_{0:k-1}) = \mathcal{N}(\mathbf{x}_{k|k-1}; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) \quad (2.10b)$$

The KF equations which are calculated based on (2.5), (2.7), and equations above are given below. Note that, in the sequel, the predicted and posterior statistics will have $_{k|k-1}$ and $_{k|k}$ as subscripts, respectively.

- Prediction Update Equations:

$$\mathbf{m}_{k|k-1} = \mathbf{F}\mathbf{m}_{k-1|k-1}, \quad (2.11a)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{m}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q}. \quad (2.11b)$$

- Measurement Update Equations:

$$\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}, \quad (2.12a)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}_k^{-1}, \quad (2.12b)$$

$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k(y_k - \mathbf{H}\mathbf{m}_{k|k-1}), \quad (2.12c)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}\mathbf{P}_{k|k-1}. \quad (2.12d)$$

In the first two equations of (2.12), the calculated variables are called innovation matrix $\mathbf{S}_k$, and the Kalman gain $\mathbf{K}_k$. Based on the equations given in (2.11) and (2.12), we can estimate the state variables recursively.

2.4 Bayesian Conjugacy

With the Kalman filter, we achieved our goal and obtained some compact formulas for Gaussian distributed densities. In the Bayesian approach, the posterior distribution is proportional to the product of the prior distribution and the likelihood function (2.2). In most applications, the posterior and prior distributions are desired to be in the same probability distribution family. If this is satisfied, the prior is then called the *conjugate prior* for the likelihood function [56]. In the KF, since we are performing linear operations with Gaussian distributions, our posterior distribution is also a Gaussian distribution. This property of the Gaussian distributions is named *self-conjugacy*.

There are various conjugate pairs in the literature [56]. In this thesis, in addition to the Gaussian self-conjugacy, we will focus on;

- The conjugacy is defined by the inverse-Wishart (IW) distribution for the positive semi-definite covariance matrix of a multivariate Gaussian distribution.

- The conjugacy is defined by the inverse-Gamma (IG) distribution for the positive variance of a Gaussian distribution.

In the RMM based ETT, literature researchers take advantage of these utilities in order to define appropriate distributions for the extent of the target.

2.5 Variational Bayes Approach

In most probabilistic approaches, a posterior density is required to be calculated. However, in some cases, it might not be feasible to achieve that. In these situations, one has to approximate the posterior density rather than calculating exactly. In this manner, a distance metric between the true posterior density, $p(\cdot)$, and the approximation, $q(\cdot)$, has to be defined and minimized in order to achieve a desirable result. A possible distance metric to use is the Kullback-Leibler (KL) divergence 2.13. In this thesis, we utilized the variational Bayes approach [8, Ch. 10], which incorporates a recursive solution to the optimization problem by minimizing the KL divergence to find the optimal approximate density in a specified family of distributions $\mathcal{M}$.

$$q^*(\cdot) = \underset{q(\cdot) \in \mathcal{M}}{\operatorname{argmin}} \operatorname{KL}(q(\cdot) || p(\cdot)) = \underset{q(\cdot) \in \mathcal{M}}{\operatorname{argmin}} \int q(\cdot) \log \left[\frac{q(\cdot)}{p(\cdot)} \right] \quad (2.13)$$

The KL divergence formula is analyzed below for two exemplary probability distributions.

$$\begin{aligned} \operatorname{KL}(q(z) || p(z|x)) &= \int_z q(z) \log \left[\frac{q(z)}{p(z|x)} \right] dz \\ &= \int_z [q(z) \log q(z)] dz - \int_z [q(z) \log p(z|x)] dz \\ &= \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(z|x)] \\ &= \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log \left[\frac{p(x, z)}{p(x)} \right]] \\ &= \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(x, z)] + \mathbb{E}_q[\log p(x)] \\ &= \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(x, z)] + \log p(x) \end{aligned} \quad (2.14)$$

where $p(z|x)$, and $q(z)$ are the posterior distribution, and the approximation, respectively. Since the KL divergence includes a term $\log p(x)$ which is not related with $q(\cdot)$, it is not tractable. However, since $\log p(x)$ is a constant we can optimize the KL divergence without considering it. Note that, KL divergence is not symmetric, i.e., $\text{KL}(q(z)||p(z|x)) \neq \text{KL}(p(z|x)||q(z))$. One can verify this property by inspecting the terms in 2.14.

2.5.1 Factorized Distributions

Factorized distributions are frequently utilized to solve the variational Bayes problem [8, Ch. 10]. It also corresponds to the approximation framework called *mean field theory* in physics [65].

First of all, we reckon a set of the latent variables $\mathcal{Z}$ which consists of individual elements indicated by $\mathbf{z}_i$ where each probability distribution could be in any type and $i = 1, \dots, N$. The major assumption in this approach is

$$q(\mathcal{Z}) = \prod_{i=1}^N q_i(\mathbf{z}_i). \quad (2.15)$$

Now that we stated our assumption, it is required to find the distribution set which minimizes the KL divergence. Let's suppose that our observed variables are described by $\mathcal{X} = x_1, x_2, \dots, x_N$. The KL divergence equation will be as following.

$$\begin{aligned} \text{KL}(q(\mathcal{Z})||p(\mathcal{Z}|\mathcal{X})) &= \mathbb{E}_q[\log q(\mathcal{Z})] - \mathbb{E}_q[\log p(\mathcal{X}, \mathcal{Z})] + \log p(\mathcal{X}) \\ &= \mathcal{L}(q(\mathcal{Z})) + \log p(\mathcal{X}) \end{aligned} \quad (2.16)$$

$$\log p(\mathcal{X}) = \mathcal{L}(q(\mathcal{Z})) + \text{KL}(q(\mathcal{Z})||p(\mathcal{Z}|\mathcal{X})) \quad (2.17)$$

where $\mathcal{L}(q(\mathcal{Z})) = \mathbb{E}_q[\log q(\mathcal{Z})] - \mathbb{E}_q[\log p(\mathcal{X}, \mathcal{Z})]$. Note that, the term $\mathcal{L}(q(\mathcal{Z}))$ acts as a lower limit in (2.17) when the KL divergence goes to zero. Hence, $\mathcal{L}(q(\mathcal{Z}))$ is described as evidence lower band (2.18).

$$\begin{aligned} \log p(\mathcal{X}) &= \mathcal{L}(q(\mathcal{Z})) + \text{KL}(q(\mathcal{Z})||p(\mathcal{Z}|\mathcal{X})) \\ \log p(\mathcal{X}) &\geq \mathcal{L}(q(\mathcal{Z})) \end{aligned} \quad (2.18)$$

In this context, minimizing KL divergence equals to maximizing the evidence lower band.

$$\begin{aligned}
q^*(\mathcal{Z}) &= \operatorname{argmin}_{q(\cdot) \in \mathcal{M}} \operatorname{KL}(q(\mathcal{Z}) || p(\mathcal{Z} | \mathcal{X})) \\
&= \operatorname{argmax}_{q(\cdot) \in \mathcal{M}} \mathcal{L}(q(\mathcal{Z})) \\
&= \operatorname{argmax}_{q(\cdot) \in \mathcal{M}} \left[\mathbb{E}_q[\log q(\mathcal{Z})] - \mathbb{E}_q[\log p(\mathcal{X}, \mathcal{Z})] \right] \tag{2.19}
\end{aligned}$$

From now on, we need to find the particular set of distributions $q^*(\mathcal{Z})$, which will maximize the evidence lower band. We achieve this by optimizing each distribution individually with respect to the other distributions in the set. The derivation of a single component $q_j^*(\mathbf{z}_j)$ is given below.

$$\begin{aligned}
\mathcal{L}(q) &= \int \prod_i q_i(\mathbf{z}_i) \left\{ \log p(\mathcal{X}, \mathcal{Z}) - \sum_i \log q_i(\mathbf{z}_i) \right\} d\mathcal{Z} \\
&= \int q_j(\mathbf{z}_j) \left\{ \int \log p(\mathcal{X}, \mathcal{Z}) \prod_{i \neq j} q_i(\mathbf{z}_i) d\mathbf{z}_i \right\} d\mathbf{z}_j - \int q_j(\mathbf{z}_j) \log q_j(\mathbf{z}_j) d\mathbf{z}_j + c \\
&= \int q_j(\mathbf{z}_j) \log \hat{p}(\mathcal{X}, \mathbf{z}_j) d\mathbf{z}_j - \int q_j(\mathbf{z}_j) \log q_j(\mathbf{z}_j) d\mathbf{z}_j + c \\
&= \int q_j(\mathbf{z}_j) \frac{\log \hat{p}(\mathcal{X}, \mathbf{z}_j)}{\log q_j(\mathbf{z}_j)} d\mathbf{z}_j + c \\
&= -\operatorname{KL}(q(\mathcal{Z}) || \hat{p}(\mathcal{X}, \mathbf{z}_j)) \tag{2.20}
\end{aligned}$$

where

$$\begin{aligned}
\log \hat{p}(\mathcal{X}, \mathbf{z}_j) &= \left\{ \int \log p(\mathcal{X}, \mathcal{Z}) \prod_{i \neq j} q_i d\mathbf{z}_i \right\} \\
&= \mathbb{E}_{i \neq j}[\log p(\mathcal{X}, \mathcal{Z})] + c. \tag{2.21}
\end{aligned}$$

By looking at 2.20, we note that maximizing the lower bound is equal to minimizing the KL divergence between $q(\mathcal{Z})$ and $\hat{p}(\mathcal{X}, \mathbf{z}_j)$. Since KL divergence is a strictly positive function, the minimum occurs when $q(\mathcal{Z}) = \hat{p}(\mathcal{X}, \mathbf{z}_j)$. Therefore;

$$\log q_j^*(\mathbf{z}_j) = \mathbb{E}_{i \neq j}[\log p(\mathcal{X}, \mathcal{Z})] + c. \tag{2.22}$$

In conclusion, the logarithm of the optimal solution for a particular distribution q_j is obtained by keeping other factors still; and evaluating the expected logarithmic value

of the joint distribution of hidden and latent variables with respect to all elements of $\mathcal{Z}$ except the one that is being optimized.

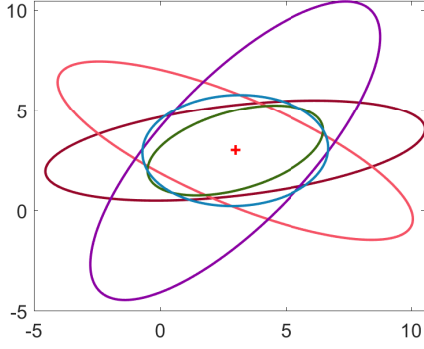
2.6 Random Matrix Model

In ETT, there are a plethora of different approaches. The earliest methods consider a predefined shape, such as a line, a circle, or a rectangle, in order to estimate the extent of the target. Gaussian Process (GP) based ETT algorithms, and random hyper-surface models (RHM) algorithms are among the most used algorithms to predict the shape of targets with more complex extents. In GP-ETT approaches [43, 82], the target's contour is defined as an unknown radial function with a GP prior. On the other hand, in RHM [4, 5, 90] it is assumed that measurements are originated from an unknown random surface.

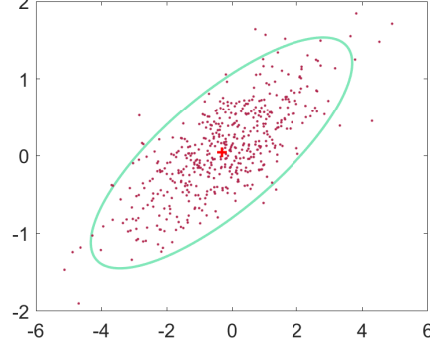
The random matrix model (RMM) [18, 42, 44, 60], which was initiated by Koch [42], has become one of the most researched areas of ETT in recent years. In essence, the RMM is introduced as estimating the extent of any target with an ellipse while tracking the kinematic properties. The ellipse is defined with an SPD matrix, $\mathbf{X}$, which can be originated from several matrix variate distributions [30]. The dimension of $\mathbf{X}$ represents the dimension of object's extent representation, i.e., if $\mathbf{X} \in \mathbb{R}^{2 \times 2}$, the extent is represented by a 2D plot. Since $\mathbf{X}$ is a SPD matrix, the implied object shape would be an ellipse. In tracking applications, where the measurements are assumed to be Gaussian, the Gaussian and the inverse Wishart distributions specify conjugate priors for the unknown mean and the covariance of the measurement density. This led the researchers to define the target's extent with the inverse Wishart distribution [42].

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{X}_k | \mathcal{Y}_k) &= p(\mathbf{x}_k | \mathbf{X}_k, \mathcal{Y}_k) p(\mathbf{X}_k | \mathcal{Y}_k) \\ &= \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k}, \mathbf{P}_{k|k} \otimes \mathbf{X}_k) \mathcal{IW}(\mathbf{X}_k; v_{k|k}, \mathbf{V}_{k|k}) \end{aligned} \quad (2.23)$$

where the kinematic vector, $\mathbf{x}_k$, is defined to be a Gaussian vector with $\mathbf{m}_{k|k}$, and $\mathbf{P}_{k|k} \otimes \mathbf{X}_k$ as the mean and the covariance. The sufficient statistics for the inverse Wishart density are the degrees of freedom, $v_{k|k}$, and the scale matrix $\mathbf{V}_{k|k}$. Vi-



(a) Samples gathered from the inverse Wishart distribution $\mathcal{IW}(\mathbf{X}; 3, \begin{bmatrix} 10 & 3 \\ 3 & 2 \end{bmatrix})$



(b) Measurements generated from a Gaussian distribution with unknown mean and unknown covariance $\mathcal{N}(\mathbf{y}; \mathbf{x}, \mathbf{X})$ whose conjugate prior distributions are set to $\mathcal{N}(\mathbf{x}; \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix})$, and

$$\mathcal{IW}(\mathbf{X}; 3, \begin{bmatrix} 10 & 3 \\ 3 & 2 \end{bmatrix})$$

Figure 2.1: Inverse Wishart distribution sampling and measurements visualization examples

visualizations of the samples taken from an exemplary inverse Wishart distribution $\mathcal{IW}(\mathbf{X}; 3, \begin{bmatrix} 10 & 3 \\ 3 & 2 \end{bmatrix})$ with a specific center location $\begin{bmatrix} 3 \\ 3 \end{bmatrix}$, and measurements generated from a Gaussian distribution with unknown mean and unknown covariance $\mathcal{N}(\mathbf{y}; \mathbf{x}, \mathbf{X})$ whose conjugate prior distributions are set to $\mathcal{N}(\mathbf{x}; \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix})$, and $\mathcal{IW}(\mathbf{X}; 3, \begin{bmatrix} 10 & 3 \\ 3 & 2 \end{bmatrix})$ are depicted in Fig. 2.1a and Fig. 2.1b, respectively .

Two of the biggest challenges in RMM-based methods are that the measurement update is analytically not tractable, and the resulting posterior density does not satisfy the conjugacy. There are different approaches to solve these problems. In the pioneer work [42], the measurement noise is neglected to achieve Bayesian conjugacy. The follow-up research [18] incorporates the measurement noise in consideration of exact inference. This is later improved in [60] where a Bayesian interference is utilized in order to approximate the intractable posteriors. In these works, the changes in the extent of the observed part of the target are handled only by a forgetting factor which

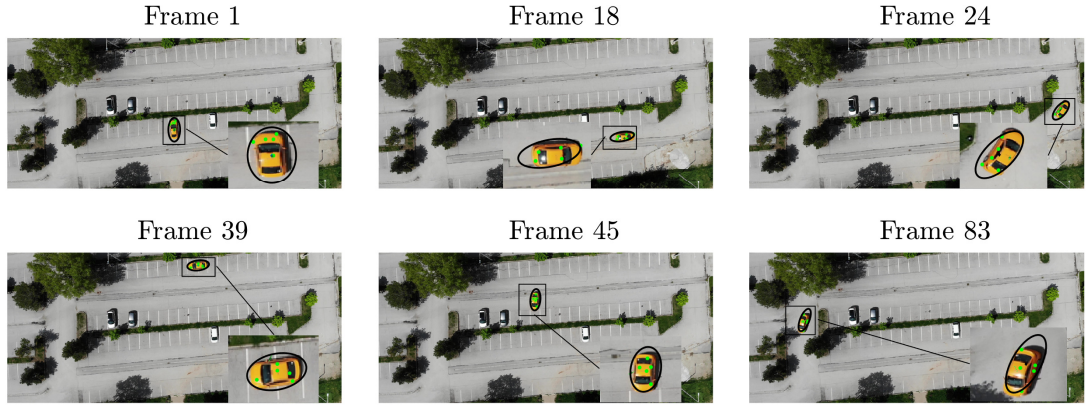


Figure 2.2: An illustration of the RM based method proposed in [78].

increases the uncertainty of the estimate. Such an approach is not troublesome as it tries to solve the problem by renouncing the information collected over the course of the tracking process. More recent RMM studies [25, 78, 88–90] focus on, also, estimating the heading-angle of the target. The target’s orientation is calculated in [25] using data from the target’s trajectory. In [88–90], the approaches provide parametric expression of the unknown extent by using an inference method that utilizes extended Kalman filters and pseudo-measurements. The method in [78] exploits the variational Bayes approach to estimate the heading-angle of the target. An exemplary visual for the RM based methods [78] is depicted in Fig. 2.2.

2.7 Machine learning with Neural Networks

2.7.1 Learning Types in Machine Learning

In recent years, we have begun to see machine learning applications being used in almost every aspect of our lives. Machine learning is actually a subtopic of the broader field of artificial intelligence and points towards the automation of various human-driven tasks. Machine learning algorithms are being utilized for different purposes such as classification, estimation, recognition, translation, prediction, etc. These methods accomplish their purposes by discovering the overt or hidden patterns that lie in the data, i.e., by learning from the data.

There are mainly three types of learning in the machine learning context. The first learning method is supervised learning. In this type of learning, the model is supplied with a unique label for each data instance. In other words, the scientist has to know the correct outcome of any input in the dataset. Therefore, the model is able to calculate the error between its predictions with the actual true value and adjust its parameters accordingly. An example of the supervised learning problem is the classification of vehicles based on some features such as the size of the vehicle, the brand of the vehicle, the mileage, the tire radius, etc. In this particular example, the training data should consist of extracted features of a large number of vehicles and their related categories. The second type of learning is the sibling of the supervised learning; unsupervised learning. In unsupervised learning, the model and often the scientist does not know the correct outcome for any given input data. Most of the time, unsupervised learning problems are considered to be more difficult than supervised learning problems. Since the analysis and labeling of big data is often an expensive process, unsupervised learning approaches are quite important. Some purposes of this kind of learning can be given as finding any patterns among the data to categorize it accordingly without requiring any external interference, feature learning for further learning algorithms, etc. Clustering the car drivers as aggressive and calm based on their driving behaviors could be given as a real-life example of the unsupervised learning problem. The last but not least type of learning is reinforcement learning. Even though reinforcement learning is similar to supervised learning, we cannot consider them the same. In reinforcement learning, a feedback is supplied to the so-called agent based on how well it performs in the given assignment. The actions of the agent are rewarded or punished according to a reward function. The main goal for the agent is to obtain the highest reward possible by following a trial-and-error procedure. After each run, the probabilities of the agent's possible actions are updated based on the reward collected. In essence, reinforcement learning strongly relates to the process of learning by humans and animals. An example of this similarity could be given as how human babies learn to walk and how agents learn to walk [32]. In this thesis, we will focus on supervised learning for neural network algorithms and discuss further in the following sections.

2.7.1.1 Supervised Learning

One of the most crucial features of neural networks, perhaps the one that still drives us to utilize them, is their high-performance learning capabilities using data. Earlier, we underlined the similarities between artificial neural networks and the biological nervous system. Another prominent similarity would be their abilities to learn. Even though we do not precisely understand how the learning process occurs in the brain, our understanding of how the neural network learns is solid.

2.7.2 Biological Interpretation of Neural Networks

In the past decades, researchers have been trying to build a computational model similar to a natural supercomputer that is our brain. Neural networks (NN) or, more formally, artificial neural networks (ANN) can be interpreted as a model that is inspired by how the human brain processes information and provides an output. From a highly simplified biological perspective, the brain consists of neurons, which are the basic computational units that manipulate data and transmit information to synapses through axons. The so-called output of the previous neuron is gathered from the previous neuron's synapses by the dendrites, and the process repeats itself. The computational model of a NN mimics the behavior of the biological model, Fig. 2.3. Similar to the biological point of view, the computational model of a NN consists of artificial neurons where the computations take place. Each artificial neuron's input is associated with a weight that represents synaptic strength. These weights are different from each other and vary according to the importance of the input signal. Lastly, weighted input data is accumulated in the neuron by summation similar to dendrites.

2.7.3 Neural Network Architecture

Similar to a brain, computational models of neural networks comprise so-called neurons. In a simple neural network architecture, neurons are combined to create three main layers: the input layer, the hidden layer, and the output layer. Nowadays, neu-

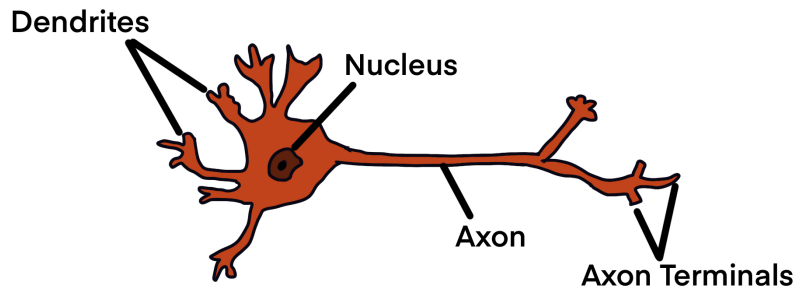


Figure 2.3: A basic biological representation of a neuron.

rons in each layer are connected to neurons in the previous and subsequent layers; however, there are no connections between neurons within the layer. An example of a primary neural network is visualized in Fig. 2.4.

2.7.3.1 Input Layer

The input layer consists of a series of neurons ordered in a single layer. In essence, it gathers the information from the data and feeds them into the network without applying any operations. The number of neurons is determined based on the number of features extracted from the data. For example, in Fig. 2.4, the number of features is set to 3.

2.7.3.2 Output Layer

The output layer is a single layer of neurons containing several neurons that is equal to the number of distinct classes. By definition, it is safe to say that there are 4 different classes in the dataset that is being used by the network in Fig. 2.4. A non-linear activation function that maps the output to the desired format is applied at the end of the output layer. There are different activation functions that will be discussed in the coming sections.

2.7.3.3 Hidden Layer

Hidden layers are where the magic of the neural network happens. In these layers, the processing of the data is accomplished. The number of hidden layers and the number of neurons in each layer depends on the size of the data, the number of features, and the structure of the data. Note that one should be careful while adjusting the number of hidden layers and the number of neurons. As the number of hidden layers increases, the more complex the network becomes. Having a model that is more complex than enough would result in overfitting the data, which will yield poor performance in the testing phase. On the other hand, a more straightforward model might underfit the data, causing the model to underperform during the testing stage, as in the case of overfitting. There are also algorithms whose purpose is to optimize these hyper-parameters, i.e., the number of layers and the number of neurons [52]. The output of a single j^{th} neuron in the $(k + 1)^{th}$ hidden layer is calculated as the following equation. A visualization of the calculation is shown in Fig. 2.5.

$$a_j^{k+1} = \sigma(b_j^{k+1} + \sum_{i=1}^n w_{j,i}^k a_i^k) \quad (2.24)$$

where

- a_i^k denotes the output of the i^{th} neuron in layer $\#k$,
- $w_{j,i}^k$ is the weight value of the connection between the i^{th} neuron in layer $\#k$ and j^{th} neuron in layer $\#(k + 1)$,
- b_j^{k+1} is called bias and defined uniquely for each neuron,
- $\sigma(\cdot)$ is the selected activation function.

2.7.3.4 Learning Mechanism

In machine learning, learning is referred to the tuning of network parameters good enough to predict an output based on an input reasonably accurate. In order to up-

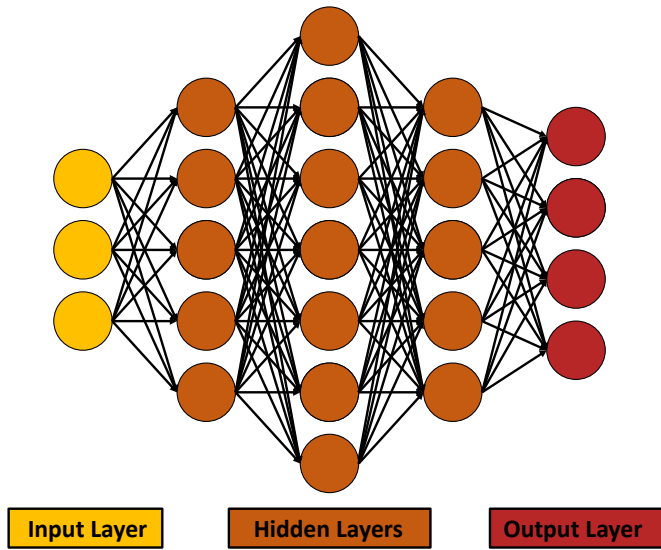


Figure 2.4: An example neural network model.

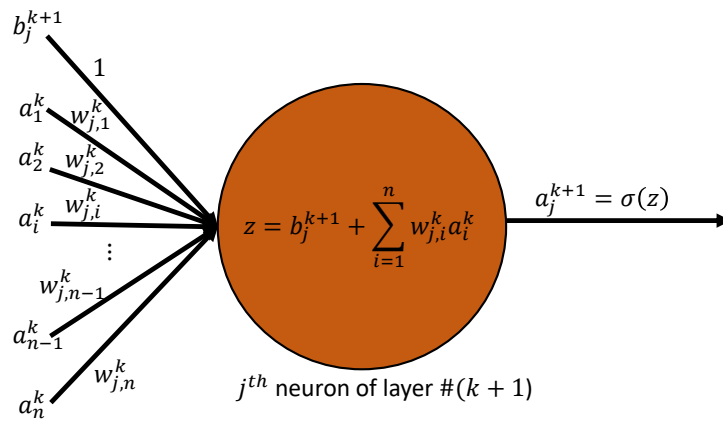


Figure 2.5: A deeper look of a single neuron in a hidden layer

date the parameters of a neural network, namely the weights and bias values, there has to be a feedback mechanism inside the network. In the literature, this feedback mechanism is called the *backpropagation algorithm* [48]. A summary of the main principles in a single cycle of the backpropagation algorithm is stated below before a broader explanation.

- Calculate the output of the network for a given input.
- Calculate the error, i.e., the loss, according to a loss function using the ground truth and the predicted value for that particular input.
- Calculate the derivatives of the loss value with respect to the loss value via chain-rule.
- Update the parameters with a product of a hyper-parameter called learning rate and the derivation.

One of the crucial elements of backpropagation algorithm is choosing a well-suited loss function. A loss function effectively calculates the error between the desired output and the network's prediction. There are a plethora of loss functions in the literature. Some of the most used ones are given below with their use-cases.

- Mean Square Error (MSE) also known as Quadratic Loss and L2 Loss

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N (y^{(i)} - f_{\theta}(x^{(i)}))^2 \quad (2.25)$$

MSE is calculated as the average of squared error between ground truths $y^{(i)}$ and predictions of the network $f_{\theta}(x^{(i)})$ which is parametrized by θ per input $x^{(i)}$. The square of the error has several benefits. The first one is that predictions that are further away from their desired values will be punished severely. Secondly, since the calculated squared error is always positive, different terms cannot cancel each other, so it does not output a meaningless result.

- Cross-Entropy Loss

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_j^{(i)} \log(p_j(f_{\theta}(x^{(i)}))) \quad (2.26)$$

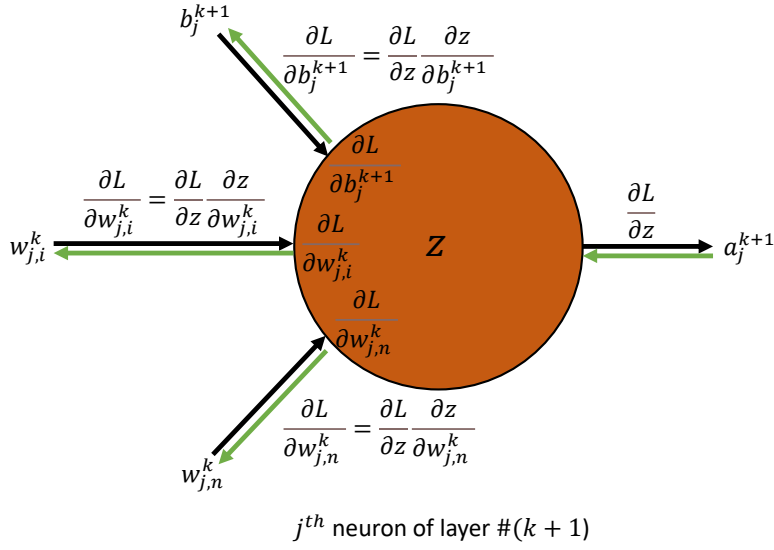


Figure 2.6: An illustration of the backpropagation algorithm.

where $p_j(f_\theta(x^{(i)}))$ is the probability of the output $f_\theta(x^{(i)})$ to belong in class- j . Cross-entropy losses are usually used for classification purposes.

After the error is calculated, it is then back-propagated to the network in order to update parameters by utilizing the chain rule. A derivation of the chain rule for a single neuron and an illustration of the backpropagation algorithm is given in Fig. 2.6.

There are different optimization techniques to update parameters. One of the most employed algorithms is the gradient-descent approach. Gradient-descent has advantages such as having low complexity and being computationally efficient. The formula of the gradient-descent approach for a single parameter is depicted below.

$$w_{j,i,updated}^k = w_{j,i}^k - \alpha \frac{\partial L(\theta)}{\partial w_{j,i}^k} \quad (2.27)$$

where α is the learning rate and needs to be chosen wisely. An α which is too large would cause the parameter to oscillate, and with a too-small α the convergence rate would be too slow.

2.8 Publications

In the last two years, the following manuscripts have been published:

Tuncer, B., Özkan, E., & Orguner, U. (2021). Multi-Ellipsoidal Extended Target Tracking with Variational Bayes Inference.

Tuncer, B., & Özkan, E. (2021). Random matrix based extended target tracking with orientation: A new model and inference. *IEEE Transactions on Signal Processing*, 69, 1910-1923.

Tuncer, B., Kumru, M., & Özkan, E. (2019, July). Extended Target Tracking and Classification Using Neural Networks. In 2019 22th International Conference on Information Fusion (FUSION) (pp. 1-7). IEEE.



CHAPTER 3

RANDOM MATRIX BASED EXTENDED TARGET TRACKING WITH ORIENTATION: A NEW MODEL AND INFERENCE

3.1 Introduction

Extended target tracking (ETT) problem involves processing multiple measurements that belong to a single target at each scan. In contrast to conventional tracking algorithms, which rely on point target assumption, ETT algorithms aim at estimating the target extent, which can be defined as the target-specific region that generates multiple measurements. Previous studies in the ETT literature can be broadly categorized into four groups:

- Simple shape models
- Random matrix (RM) based models
- Random hyper-surface (RHS) based models
- Mixture models

A simple approach to ETT involves assuming a predefined shape for the extent/contour of the object such as a circle, a rectangle, or a line [6, 23, 24]. The most common approaches in the literature utilize RM models, where the target extent is represented by an ellipse [18, 42, 44, 60]. Alternatively, RHS models are suggested in [4, 5]. More recently, Gaussian Process (GP) based models are proposed for extended target tracking [43, 63, 82]. Another fold of studies focuses on modeling the target extent with multiple ellipses [21, 34, 40, 45].

RM models represent the elliptical extent of a target by an unknown positive semi-definite matrix (PSDM). In the Bayesian framework, inverse-Wishart (IW) distribution defines a conjugate prior for PSDMs. In RM based ETT models, the overall target state is composed of a Gaussian kinematic state vector and an IW distributed extent matrix. Several algorithms are proposed to approximate or compute the posterior of this augmented state. In [42], exact inference is performed by neglecting the measurement noise and exploiting the resulting conjugacy. This model is restrictive in the sense that the kinematic state vector has to be composed of the target's position and higher-order spatial components such as velocity and acceleration. Koch's RM model is later improved in [18] to account for the measurement noise in the updates at the expense of exact inference. The update equations in [18] are intuitive, but the approximations are difficult to quantify theoretically. This problem is later addressed by [60], where the variational Bayes technique is used to obtain approximate posteriors.

None of the aforementioned RM models is capable of tracking the heading angle of an extended target. They instead rely on a forgetting factor to forget the sufficient statistics of the unknown extent matrix in time to account for the changes in the orientation of the target. Such an approach is problematic as it aims to discard the information collected in the past and try to explain the change in the orientation as the change in the target shape. There are earlier studies that aim at estimating the orientation angle of elliptical objects [25, 88–90]. In [25], the orientation of the target is estimated by using the information that is obtained from the trajectory of the target. The methods that are proposed in [88–90] express the unknown extent parametrically and perform inference using extended Kalman filters together with pseudo-measurements. In these approaches, an explicit nonlinear measurement equation is derived where the kinematic and shape parameters are related to measurements by multiplicative random variables. The inference in [88] involves second-order Taylor series approximation to approximate the pseudo-measurement covariance matrix. In [89], the authors improved the algorithm in [88] further and eliminated the need for computing Hessian matrices. Instead, they showed that the expectation and the covariance of the pseudo-measurements could be approximated from the original measurement covari-

ance matrix. In [90], the predicted measurement covariance matrix approximation is calculated more precisely.

There are several drawbacks of the methods in [88–90]. The models used in these methods involve a multiplicative noise term, which introduces additional non-linearity in the problem, and it makes performing inference more difficult. The methods require a pseudo-measurement, which must be constructed from the original measurements to update kinematic and extent states separately. The measurements collected at one time instant must be processed sequentially. Changing the order of the measurements causes minor changes in the performance [90]. The state variables corresponding to the semi-axes lengths, which are positive by definition, are distributed with Gaussian distributions whose support covers both positive and negative real line. In some cases, it can be challenging to reflect available information into the priors defined in [90], which may cause a collapse in the extent estimates in the subsequent measurement updates.

In this chapter, we propose a novel RM model that defines a Gaussian prior for the heading angle and an inverse Gamma prior for the extent parameters, which guarantee positive semi-definiteness. It is not possible to find a closed-form expression for the resulting posterior hence we utilize the variational Bayes technique to perform approximate inference. The variational Bayes technique is successfully applied to complex filtering problems in the literature to obtain approximate posteriors [35, 58–60, 71, 74].

The contributions of this chapter can be listed as follows.

- We provide a novel solution that can track the orientation of a target and estimate its extent jointly.
- The proposed solution utilizes appropriate priors, which are defined over non-negative real numbers, for the unknown extent parameters.
- The problem formulation does not rely on multiplicative noise terms or pseudo-measurements.

- The measurement update can be performed by processing multiple measurements as a batch. The update does not depend on the order of the measurements.
- The uncertainty in the orientation and shape parameters can be expressed separately.
- The inference is performed via the well-known variational Bayes technique.
- The update equations are easy to implement, and the algorithm can be used in real-time tracking applications. We illustrate the performance of the method in simulations and experiments with real data.
- The proposed method outperforms the state-of-the-art methods when compared with respect to accuracy and robustness.

The following sections are organized as follows. In Section 3.2, we formulate the problem of joint shape estimation and tracking of elliptical objects with time-varying orientation. In the subsequent sections, we present the inference method. The measurement update is derived in Section 3.3. Time update is given in Section 3.4. A closer look at a single measurement update and its comparison with the state-of-the-art extended Kalman filter (EKF) algorithm is given in Section 3.5. Lastly, the results are presented and discussed in Section 3.6.

Table 3.1: Notations

-
- Set of real matrices of size $m \times n$ is represented with $\mathbb{R}^{m \times n}$.
 - Set of symmetric positive definite and semi-definite matrices of size $n \times n$ is represented with $\mathbb{S}_{++}^n$ and $\mathbb{S}_+^n$, respectively.
 - $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \Sigma)$ represents the multivariate Gaussian distributions with mean vector $\boldsymbol{\mu} \in \mathbb{R}^{n_x}$ and covariance matrix $\Sigma \in \mathbb{S}_{++}^{n_x}$,
 - $\mathcal{IG}(\sigma; \alpha, \beta)$ represents the inverse Gamma distribution over the scalar $\sigma \in \mathbb{R}^+$ with shape and scale parameters $\alpha \in \mathbb{R}^+$ and $\beta \in \mathbb{R}^+$ respectively,

$$\mathcal{IG}(\sigma; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \sigma^{-\alpha-1} \exp\left(-\frac{\beta}{\sigma}\right),$$

- The number of measurements at time k is represented by $m_k \in \mathbb{Z}^+$.
- For given measurement number of m_k , $\mathcal{Y}_k$ represents the measurement set $\{\mathbf{y}_k^1, \dots, \mathbf{y}_k^{m_k}\}$ at time k .
- For any number $a \in \mathbb{Z}^+$, $\mathcal{Z}_k$ represents the variable set $\{\mathbf{z}_k^1, \dots, \mathbf{z}_k^a\}$ at time k .
- $\mathbf{r}_k$ represent the vector $[r_k^1, \dots, r_k^a]^T$ with size $a \in \mathbb{Z}^+$.
- KL denotes the Kullback-Leibler divergence between two distributions $q(x)$ and $p(x)$,

$$\text{KL}(q(x)||p(x)) \triangleq \int q(x) \log\left(\frac{q(x)}{p(x)}\right) dx.$$

- $\det(\mathbf{A})$ denotes the determinant of matrix $\mathbf{A}$.
-

Table 3.1: Notations (continued)

-
- $\text{Tr} [\mathbf{A}] = \sum_{i=1}^n a_{ii}$ where a_{ii} is the i^{th} diagonal element of $\mathbf{A} \in \mathbb{R}^{n \times n}$.
 - $\mathbb{E}_p$ denotes the expectation operator, and p emphasizes the underlying probability distribution(s).
 - $\text{diag}(a_1, a_2, \dots, a_n)$ returns the diagonal matrix whose diagonal elements are $a_1, a_2, \dots, a_n$.
 - $\text{blkdiag}(\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_N)$ returns the block diagonal square matrix whose main-diagonal blocks are the input matrices $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_N$.
 - *h.o.t.* stands for higher-order terms.
-

3.2 Problem Definition

Consider a single target from which multiple measurements are generated in a single scan. Assume that the state of the extended target consists of the kinematic state $\mathbf{x}_k \in \mathbb{R}^{n_x}$, the orientation angle $\theta_k \in \mathbb{R}$, and the diagonal positive definite target extent matrix $\mathbf{X}_k \in \mathbb{R}^{n_y \times n_y}$, $\mathbf{X}_k \triangleq \text{diag}(\sigma_k^1, \sigma_k^2, \dots, \sigma_k^{n_y})$, where n_x and n_y represent the dimensions of the kinematic target state and the measurements, respectively. Given $\mathbf{x}_k, \mathbf{X}_k$ and θ_k , the measurements generated by the target are assumed to be independent and identically distributed,

$$p(\mathbf{y}_k^j | \mathbf{x}_k, \mathbf{X}_k, \theta_k) \sim \mathcal{N}(\mathbf{y}_k^j; \mathbf{H}\mathbf{x}_k, s\mathbf{T}_{\theta_k}\mathbf{X}_k\mathbf{T}_{\theta_k}^T + \mathbf{R}), \quad (3.1)$$

where

- $\mathbf{y}_k^j \in \mathbb{R}^{n_y}$ is the j^{th} measurement at time k ,
- $\mathbf{H} \in \mathbb{R}^{n_y \times n_x}$ is the measurement matrix,
- $\mathbf{R} \in \mathbb{R}^{n_y \times n_y}$ is the positive definite measurement noise covariance matrix,
- $s \in \mathbb{R}^+$ is the scaling parameter,

- $\mathbf{T}_{\theta_k} \in \mathbb{R}^{n_y \times n_y}$ is the rotation matrix which performs a rotation around the center of the target by the orientation angle θ_k . $\mathbf{T}_{\theta_k}$ satisfies the well known properties of the rotation matrices such as $\mathbf{T}_{\theta_k}^{-1} = \mathbf{T}_{\theta_k}^T$, and $\det(\mathbf{T}_{\theta_k}) = 1$. In 2D, it is defined as,

$$\mathbf{T}_{\theta_k} \triangleq \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{bmatrix}. \quad (3.2)$$

Note that the measurement likelihood in (3.1) can be interpreted as a measurement model with two additive Gaussian terms; one with time-varying, unknown but state-dependent statistics, $v_k^I(\mathbf{X}_k, \theta_k) \sim \mathcal{N}(0, s\mathbf{T}_{\theta_k}\mathbf{X}_k\mathbf{T}_{\theta_k}^T)$, and one with known statistics, $v_k^{II} \sim \mathcal{N}(0, \mathbf{R})$.

$$y_k = \mathbf{H}x_k + v_k^I(\mathbf{X}_k, \theta_k) + v_k^{II} \quad (3.3)$$

The effective covariance matrix in likelihood (3.3) is unknown, time-varying, and state-dependent, which casts the main difficulty in the ETT problem together with the absence of conjugacy¹. An illustration of the resulting extent model is depicted in Figure 3.1. Similar elliptical models are frequently used in target tracking applications for tracking vehicles, vessels, pedestrians, animals, or groups of objects [26, 29, 72, 80, 81].

In the Bayesian filtering framework, we aim at estimating the unknown variables $\mathbf{x}_k, \theta_k$, and $\mathbf{X}_k$ given the measurements collected up to and including time k . To achieve this, we define appropriate priors for the unknowns and try to compute their posteriors in a recursive manner. This is generally performed by repeating two recursive steps:

- **Time Update (Prediction):** At any time step k , the predictive distribution $p(\mathbf{x}_k, \mathbf{X}_k, \theta_k | \mathcal{Y}_{1:k-1})$ is computed according to Chapman-Kolmogorov equation by using the posterior from the previous time step $k-1$, and the transition density induced by the system dynamics.
- **Measurement Update (Correction):** When the new measurements $\mathcal{Y}_k$ are available, the posterior distribution $p(\mathbf{x}_k, \mathbf{X}_k, \theta_k | \mathcal{Y}_{1:k})$ is computed by using

¹ A family of prior distributions is conjugate to a particular likelihood function if the posterior distribution belongs to the same family as the prior.

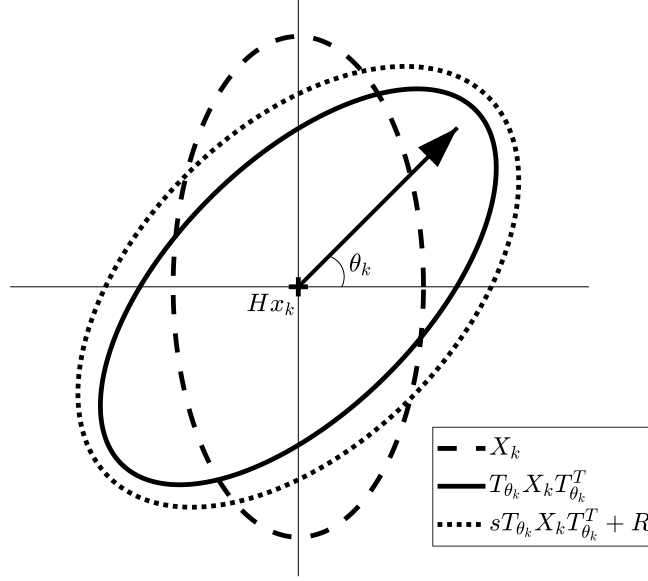


Figure 3.1: An illustration of the target extent model with ellipsoids corresponding to the covariance matrices $\mathbf{X}_k$ (dashed line), $\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T$ (solid line) and $s \mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T + \mathbf{R}$ (dotted line), respectively.

the Bayes' rule. In this step, the predictive distribution $p(\mathbf{x}_k, \mathbf{X}_k, \theta_k | \mathcal{Y}_{1:k-1})$ is used as the prior.

Unfortunately, it is not possible to obtain a closed form expression for the posterior in our problem. Therefore, we will look for an approximate analytical solution using a variational approximation.

Before introducing the details of this approximation, we will first define the prior distributions of the unknown variables. The joint prior distribution of the kinematic state, the extent, and the orientation is specified as

$$p(\mathbf{x}_0, \mathbf{X}_0, \theta_0) = \mathcal{N}(\mathbf{x}_0; \hat{\mathbf{x}}_0, P_0) \times \prod_{i=1}^{n_y} \mathcal{IG}(\sigma_0^i; \alpha_0^i, \beta_0^i) \times \mathcal{N}(\theta_0; \hat{\theta}_0, \Theta_0), \quad (3.4)$$

where $\mathbf{X}_0 \triangleq \text{diag}(\sigma_0^1, \sigma_0^2, \dots, \sigma_0^{n_y})$, and $\mathcal{IG}(\sigma_0^i; \alpha_0^i, \beta_0^i)$ denotes the inverse Gamma distribution. Here, $\hat{\mathbf{x}}_0$ and P_0 are the prior mean and covariance matrix of the Gaussian kinematic state vector $\hat{\mathbf{x}}_0$, respectively. The prior mean and covariance matrix of the orientation angle θ_0 are denoted by $\hat{\theta}_0$ and Θ_0 , respectively. Please see Table 3.1 for

the complete list of the notations.

In the following sections, we will describe the *Measurement Update* and *Time Update* steps of the proposed method in detail.

3.3 Measurement Update

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

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{X}_k, \theta_k | \mathcal{Y}_{1:k-1}) &= \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \\ &\times \prod_{i=1}^{n_y} \mathcal{IG}(\sigma_{k|k-1}^i; \alpha_{k|k-1}^i, \beta_{k|k-1}^i) \\ &\times \mathcal{N}(\theta_k; \hat{\theta}_{k|k-1}, \Theta_{k|k-1}). \end{aligned} \quad (3.5)$$

The left-hand side of the above expression is conditioned on the measurements up to and including time instant $k-1$. The predicted mean and covariance of the Gaussian state vector is represented by $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$, respectively. The shape and the scale variables for the i^{th} diagonal element of the inverse Gamma distributed extent state $\mathbf{X}_k$ are $\alpha_{k|k-1}^i$ and $\beta_{k|k-1}^i$.

When the measurements $\mathcal{Y}_k$ are available at time k , the posterior distribution can be computed using Bayes' rule

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{X}_k, \theta_k | \mathcal{Y}_{1:k}) \\ = \frac{p(\mathcal{Y}_k | \mathbf{x}_k, \mathbf{X}_k, \theta_k) p(\mathbf{x}_k, \mathbf{X}_k, \theta_k | \mathcal{Y}_{1:k-1})}{p(\mathcal{Y}_k | \mathcal{Y}_{1:k-1})}. \end{aligned} \quad (3.6)$$

By assuming conditional independence of the measurements at time k , the measurement likelihood can be factorized as

$$\begin{aligned} p(\mathcal{Y}_k | \mathbf{x}_k, \mathbf{X}_k, \theta_k) &= \prod_{j=1}^{m_k} p(\mathbf{y}_k^j | \mathbf{x}_k, \mathbf{X}_k, \theta_k) \\ &= \prod_{j=1}^{m_k} \mathcal{N}(\mathbf{y}_k^j; \mathbf{H}\mathbf{x}_k, s\mathbf{T}_{\theta_k}\mathbf{X}_k\mathbf{T}_{\theta_k}^T + \mathbf{R}). \end{aligned} \quad (3.7)$$

In the following, we will describe a variational inference based approximation method to estimate the posterior distribution using the likelihood function in (3.7).

3.3.1 Variational Inference

An approximate analytical solution for the posterior density in (3.6) can be obtained as a product of factorized probability density functions (PDFs) using a variational approximation. Before we present the details, we need to define additional instrumental variables to address the absence of conjugacy caused by the additive measurement noise covariance term R in the likelihood. We will call these variables noise-free measurements [60], and denote them with $\mathcal{Z}_k = \{\mathbf{z}_k^j\}_{j=1}^{m_k}$. By using $\mathcal{Z}_k$, the measurement likelihood in (3.7) can be expressed as

$$\begin{aligned} \mathcal{N}(\mathbf{y}_k^j; \mathbf{H}\mathbf{x}_k, s\mathbf{T}_{\theta_k}\mathbf{X}_k\mathbf{T}_{\theta_k}^T + \mathbf{R}) = \\ \int \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R})\mathcal{N}(\mathbf{z}_k^j; \mathbf{H}\mathbf{x}_k, s\mathbf{T}_{\theta_k}\mathbf{X}_k\mathbf{T}_{\theta_k}^T) d\mathbf{z}_k^j \end{aligned} \quad (3.8)$$

for a single measurement. Note that the measurement likelihood is the marginal of the following joint density

$$\begin{aligned} p(\mathbf{y}_k^j, \mathbf{z}_k^j | \mathbf{x}_k, \mathbf{X}_k) = \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) \\ \times \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}\mathbf{x}_k, s\mathbf{T}_{\theta_k}\mathbf{X}_k\mathbf{T}_{\theta_k}^T). \end{aligned} \quad (3.9)$$

Let us include the instrumental variable $\mathcal{Z}_k$ in the posterior. Later, it will be marginalized out to obtain the posterior of the states

$$p(\mathbf{x}_k, \mathbf{X}_k, \theta_k, \mathcal{Z}_k | \mathcal{Y}_{1:k}) \approx q_{\mathbf{x}}(\mathbf{x}_k)q_X(\mathbf{X}_k)q_{\theta}(\theta_k)q_{\mathcal{Z}}(\mathcal{Z}_k). \quad (3.10)$$

Here, $q_{\mathcal{Z}}(\mathcal{Z}_k)$ denotes the approximate density of the instrumental variable $\mathcal{Z}_k$. The idea of variational approximation is to seek factorized densities whose product minimizes the following cost function.

$$\begin{aligned} \hat{q}_{\mathbf{x}}, \hat{q}_X, \hat{q}_{\theta}, \hat{q}_{\mathcal{Z}} = \arg \min_{q_{\mathbf{x}}, q_X, q_{\theta}, q_{\mathcal{Z}}} \text{KL}(q_{\mathbf{x}}(\mathbf{x}_k)q_X(\mathbf{X}_k)q_{\theta}(\theta_k)q_{\mathcal{Z}}(\mathcal{Z}_k) \\ || p(\mathbf{x}_k, \mathbf{X}_k, \theta_k, \mathcal{Z}_k | \mathcal{Y}_{1:k})). \end{aligned} \quad (3.11)$$

The solution of the optimization problem (3.11) satisfies the following equation [8, Ch. 10]:

$$\log \hat{q}_\phi(\phi_k) = \mathbb{E}_{\setminus\phi} [\log p(\mathbf{x}_k, \mathbf{X}_k, \theta_k, \mathbf{Z}_k, \mathbf{y}_k | \mathbf{y}_{1:k-1})] + c_{\setminus\phi} \quad (3.12)$$

where $\phi \in \{\mathbf{x}_k, \mathbf{X}_k, \theta_k, \mathbf{Z}_k\}$, and $\setminus\phi$ is the set of all elements except ϕ , e.g., $\mathbb{E}_{\setminus\mathbf{x}_k}$ will denote expectation with respect to variables $\mathbf{X}_k, \theta_k, \mathbf{Z}_k$. The constant term with respect to variable ϕ will be denoted by $c_{\setminus\phi}$. The joint density $p(\mathbf{x}_k, \mathbf{X}_k, \theta_k, \mathbf{Z}_k, \mathbf{y}_k | \mathbf{y}_{1:k-1})$ in (3.12) can be written explicitly as

$$\begin{aligned} & p(\mathbf{x}_k, \mathbf{X}_k, \theta_k, \mathbf{Z}_k, \mathbf{y}_k | \mathbf{y}_{1:k-1}) \\ &= p(\mathbf{y}_k | \mathbf{Z}_k) p(\mathbf{Z}_k | \mathbf{x}_k, \mathbf{X}_k, \theta_k) p(\mathbf{x}_k, \mathbf{X}_k, \theta_k | \mathbf{y}_{1:k-1}) \\ &= \left(\prod_{j=1}^{m_k} \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) \right) \left(\prod_{j=1}^{m_k} \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}\mathbf{x}_k, s\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T) \right) \\ & \quad \times \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \prod_{i=1}^{n_y} \mathcal{IG}(\sigma_{k|k-1}^i; \alpha_{k|k-1}^i, \beta_{k|k-1}^i) \\ & \quad \times \mathcal{N}(\theta_k; \hat{\theta}_{k|k-1}, \Theta_{k|k-1}). \end{aligned} \quad (3.13)$$

The optimization problem (3.11) can be solved by fixed-point iterations [8, Ch. 10]. Each iteration is performed by updating only one factorized density in (3.10) while keeping all other densities fixed to their last estimated values. The update equations of the approximate densities in the $(\ell + 1)^{\text{th}}$ iteration will be given in the following subsections. To simplify the notations, $p(\mathbf{x}_k, \mathbf{X}_k, \theta_k, \mathbf{Z}_k, \mathbf{y}_k | \mathbf{y}_{1:k-1})$ is denoted as $P_{\mathbf{x}, \mathbf{X}, \theta, \mathbf{Z}, \mathbf{y}}^k$ in the sequel.

3.3.2 Calculation of $q_{\mathbf{x}}^{(\ell+1)}(\cdot)$

Substituting the previous estimates of the factorized densities into equation (3.12) yields

$$\log q_{\mathbf{x}}^{(\ell+1)}(\mathbf{x}_k) = \mathbb{E}_{\setminus\mathbf{x}_k} [\log P_{\mathbf{x}, \mathbf{X}, \theta, \mathbf{Z}, \mathbf{y}}^k] + c_{\setminus\mathbf{x}_k}. \quad (3.14)$$

The expectation above can be simplified as

$$\mathbb{E}_{\setminus\mathbf{x}_k} [\log P_{\mathbf{x}, \mathbf{X}, \theta, \mathbf{Z}, \mathbf{y}}^k]$$

$$\begin{aligned}
&= \mathbb{E}_{\mathbf{x}_k} [\log P(\mathbf{Z}_k | \mathbf{x}_k, \mathbf{X}_k, \theta_k)] \\
&\quad + \log \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\mathbf{x}_k}
\end{aligned} \tag{3.15a}$$

$$\begin{aligned}
&= \sum_{j=1}^{m_k} -0.5 \operatorname{Tr} \left[(\bar{\mathbf{z}}_k^j - \mathbf{H}\mathbf{x}_k) (\bar{\mathbf{z}}_k^j - \mathbf{H}\mathbf{x}_k)^T \right. \\
&\quad \times \mathbb{E}_{q_X^{(\ell)}, q_\theta^{(\ell)}} [(s\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T)^{-1}] \left. \right] \\
&\quad + \log \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\mathbf{x}_k}
\end{aligned} \tag{3.15b}$$

$$\begin{aligned}
&= -0.5 \operatorname{Tr} \left[m_k (\bar{\mathbf{z}}_k - \mathbf{H}\mathbf{x}_k) (\bar{\mathbf{z}}_k - \mathbf{H}\mathbf{x}_k)^T \right. \\
&\quad \times \mathbb{E}_{q_X^{(\ell)}, q_\theta^{(\ell)}} [(s\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T)^{-1}] \left. \right] \\
&\quad + \log \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\mathbf{x}_k}
\end{aligned} \tag{3.15c}$$

$$\begin{aligned}
&= \log \mathcal{N}(\bar{\mathbf{z}}_k; \mathbf{H}\mathbf{x}_k, \frac{\mathbb{E}_{q_X^{(\ell)}, q_\theta^{(\ell)}} [(s\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T)^{-1}]^{-1}}{m_k}) \\
&\quad + \log \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\mathbf{x}_k},
\end{aligned} \tag{3.15d}$$

where $\bar{\mathbf{z}}_k^j \triangleq \mathbb{E}_{q_Z^{(\ell)}}[\mathbf{z}_k^j]$, and $\bar{\mathbf{z}}_k \triangleq \frac{1}{m_k} \sum_{j=1}^{m_k} \bar{\mathbf{z}}_k^j$. It can be seen from (3.15d) that $q_{\mathbf{x}}^{(\ell+1)}(\mathbf{x}_k)$ is a Gaussian PDF with mean vector $\hat{\mathbf{x}}_{k|k}^{(\ell+1)}$ and covariance $\mathbf{P}_{k|k}^{(\ell+1)}$

$$q_{\mathbf{x}}^{(\ell+1)}(\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{(\ell+1)}, \mathbf{P}_{k|k}^{(\ell+1)}), \tag{3.16}$$

where

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k}^{(\ell+1)} &= \mathbf{P}_{k|k}^{(\ell+1)} (P_{k|k-1}^{-1} \hat{\mathbf{x}}_{k|k-1} \\
&\quad + m_k \mathbf{H}^T \mathbb{E}_{q_X^{(\ell)}, q_\theta^{(\ell)}} [(s\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T)^{-1}] \bar{\mathbf{z}}_k),
\end{aligned} \tag{3.17a}$$

$$\mathbf{P}_{k|k}^{(\ell+1)} = (P_{k|k-1}^{-1} + m_k \mathbf{H}^T \mathbb{E}_{q_X^{(\ell)}, q_\theta^{(\ell)}} [(s\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T)^{-1}] \mathbf{H})^{-1}. \tag{3.17b}$$

3.3.3 Calculation of $q_{\mathbf{X}}^{(\ell+1)}(\cdot)$

Substituting the factorized densities from the previous variational iteration into equation (3.12) yields

$$\log q_{\mathbf{X}}^{(\ell+1)}(\mathbf{X}_k) = \mathbb{E}_{\mathbf{x}_k} [\log P_{\mathbf{x}, \mathbf{X}, \theta, \mathbf{Z}, \mathbf{y}}^k] + c_{\mathbf{x}_k} \tag{3.18}$$

Substituting (3.13) into (3.18) and grouping the constant terms with respect to $\mathbf{X}_k$ results in

$$\begin{aligned} & \mathbb{E}_{\setminus \mathbf{X}_k} [\log P_{\mathbf{x}, X, \theta, \mathbf{Z}, \mathbf{Y}}^k] \\ &= \mathbb{E}_{\setminus \mathbf{X}_k} [\log P(\mathbf{Z}_k | \mathbf{x}_k, \mathbf{X}_k, \theta_k)] \\ &+ \sum_{i=1}^{n_y} \log \mathcal{IG}(\sigma_{k|k-1}^i; \alpha_{k|k-1}^i, \beta_{k|k-1}^i) + c_{\setminus \mathbf{X}_k} \end{aligned} \quad (3.19a)$$

$$\begin{aligned} &= \frac{-m_k}{2} \log |s\mathbf{X}_k| \\ &- \frac{1}{2} \text{Tr} \left[\sum_{j=1}^{m_k} \mathbb{E}_{\setminus \mathbf{X}_k} [(\mathbf{z}_k^j - \mathbf{H}\mathbf{x}_k)(\mathbf{z}_k^j - \mathbf{H}\mathbf{x}_k)^T \right. \\ &\quad \left. \times (s\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T)^{-1}] \right] \\ &+ \sum_{i=1}^{n_y} \log \mathcal{IG}(\sigma_{k|k-1}^i; \alpha_{k|k-1}^i, \beta_{k|k-1}^i) + c_{\setminus \mathbf{X}_k}. \end{aligned} \quad (3.19b)$$

Consequently, the approximate posterior density q_X follows an inverse-Gamma distribution

$$q_X^{(\ell+1)}(\mathbf{X}_k) = \prod_{i=1}^{n_y} \mathcal{IG}(\sigma_{k|k}^{i,(\ell+1)}; \alpha_{k|k}^{i,(\ell+1)}, \beta_{k|k}^{i,(\ell+1)}), \quad (3.20)$$

where

$$\alpha_{k|k}^{i,(\ell+1)} = \alpha_{k|k-1}^i + 0.5m_k, \quad (3.21a)$$

$$\beta_{k|k}^{i,(\ell+1)} = \beta_{k|k-1}^i + \frac{1}{2s} \sum_{j=1}^{m_k} \mathbb{E}_{q_{\mathbf{x}}^{(\ell)}, q_{\theta}^{(\ell)}, q_{\mathbf{Z}}^{(\ell)}} [\tilde{\mathbf{z}}_k^j (\tilde{\mathbf{z}}_k^j)^T]_{ii}, \quad (3.21b)$$

and $\tilde{\mathbf{z}}_k^j \triangleq \mathbf{T}_{\theta_k}^T (\mathbf{z}_k^j - \mathbf{H}\mathbf{x}_k)$.

3.3.4 Calculation of $q_{\mathbf{Z}}^{(\ell+1)}(\cdot)$

Substituting the factorized densities from the previous variational iteration into equation (3.12) yields

$$\log q_{\mathbf{Z}}^{(\ell+1)}(\mathbf{Z}_k) = \mathbb{E}_{\setminus \mathbf{Z}_k} [\log P_{\mathbf{x}, X, \theta, \mathbf{Z}, \mathbf{Y}}^k] + c_{\setminus \mathbf{Z}_k}. \quad (3.22)$$

The expectation above can be expressed as

$$\mathbb{E}_{\setminus \mathbf{Z}_k} [\log P_{\mathbf{x}, X, \theta, \mathbf{Z}, \mathbf{Y}}^k]$$

$$\begin{aligned}
&= \mathbb{E}_{\mathbf{Z}_k} [\log P(\mathbf{Z}_k | \mathbf{x}_k, \mathbf{X}_k, \theta_k)] \\
&\quad + \sum_{j=1}^{m_k} \log \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) + c_{\mathbf{Z}_k}
\end{aligned} \tag{3.23a}$$

$$\begin{aligned}
&= \sum_{j=1}^{m_k} \frac{-1}{2} \text{Tr} \left[(\mathbf{z}_k^j - \mathbf{H}\overline{\mathbf{x}}_k)(\mathbf{z}_k^j - \mathbf{H}\overline{\mathbf{x}}_k)^T \right. \\
&\quad \times \mathbb{E}_{q_X^{(\ell)}, q_\theta^{(\ell)}} [(s\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T)^{-1}] \left. \right] \\
&\quad + \sum_{j=1}^{m_k} \log \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) + c_{\mathbf{Z}_k},
\end{aligned} \tag{3.23b}$$

where $\overline{\mathbf{x}}_k = \mathbb{E}_{q_{\mathbf{x}}^{(\ell)}}[\mathbf{x}_k]$. Update equations for the approximate posterior density $q_{\mathbf{Z}}$ in the $(\ell + 1)^{\text{th}}$ iteration are given by

$$q_{\mathbf{Z}}^{(\ell+1)}(\mathbf{Z}_k) = \prod_{j=1}^{m_k} \mathcal{N}(\mathbf{z}_k^j; \hat{\mathbf{z}}_k^{j,(\ell+1)}, \Sigma_k k^{z,(\ell+1)}), \tag{3.24}$$

where

$$\begin{aligned}
\hat{\mathbf{z}}_k^{j,(\ell+1)} &= \Sigma_k k^{z,(\ell+1)} \left(\mathbb{E}_{q_X^{(\ell)}, q_\theta^{(\ell)}} [(s\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T)^{-1}] \mathbf{H} \mathbb{E}_{q_{\mathbf{x}}^{(\ell)}}[\mathbf{x}_k] \right. \\
&\quad \left. + \mathbf{R}^{-1} \mathbf{y}_k^j \right),
\end{aligned} \tag{3.25a}$$

$$\Sigma_k k^{z,(\ell+1)} = \left(\mathbb{E}_{q_X^{(\ell)}, q_\theta^{(\ell)}} [(s\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T)^{-1}] + \mathbf{R}^{-1} \right)^{-1}. \tag{3.25b}$$

3.3.5 Calculation of $q_\theta^{(\ell+1)}(\cdot)$

The update equations for $q_\theta^{(\ell+1)}(\cdot)$ is obtained by substituting the factorized densities from the previous variational iteration into equation (3.12)

$$\log q_\theta^{(\ell+1)}(\theta_k) = \mathbb{E}_{\mathbf{Z}_k} [\log P_{\mathbf{x}, X, \theta, \mathbf{Z}, \mathbf{y}}^k] + c_{\theta_k}. \tag{3.26}$$

Substituting (3.13) into (3.26) and grouping the constant terms with respect to θ_k results in

$$\begin{aligned}
&\mathbb{E}_{\mathbf{Z}_k} [\log P_{\mathbf{x}, X, \theta, \mathbf{Z}, \mathbf{y}}^k] \\
&= \mathbb{E}_{\mathbf{Z}_k} [\log P(\mathbf{Z}_k | \mathbf{x}_k, \mathbf{X}_k, \theta_k)]
\end{aligned}$$

$$+ \log \mathcal{N}(\theta_k; \hat{\theta}_{k|k-1}, \Theta_{k|k-1}) + c_{\setminus \theta_k} \quad (3.27a)$$

$$= \frac{-1}{2} \sum_{j=1}^{m_k} \mathbb{E}_{\setminus \theta_k} \left[\text{Tr}[(\mathbf{z}_k^j - \mathbf{H}\mathbf{x}_k)(\mathbf{z}_k^j - \mathbf{H}\mathbf{x}_k)^T \right. \\ \left. \times (s\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T)^{-1}] \right] \\ + \log \mathcal{N}(\theta_k; \hat{\theta}_{k|k-1}, \Theta_{k|k-1}) + c_{\setminus \theta_k} \quad (3.27b)$$

Unfortunately, it is not possible to obtain an exact compact form PDF for $q_{\theta}^{(\ell+1)}(\theta_k)$ because of the non-linearities involved in (3.27b). To address this issue, we will make a first order approximation of the non-linear function $f(\theta_k) \triangleq \mathbf{T}_{\theta_k}^T(\mathbf{z}_k^j - \mathbf{H}\mathbf{x}_k)$ using its Taylor series expansion around $\hat{\theta}_{k|k}^{(\ell)}$,

$$f(\theta_k) = f(\hat{\theta}_{k|k}^{(\ell)}) + \nabla f(\hat{\theta}_{k|k}^{(\ell)})(\theta_k - \hat{\theta}_{k|k}^{(\ell)}) + h.o.t., \quad (3.28)$$

$$\text{where } \nabla f(\hat{\theta}_{k|k}^{(\ell)}) \triangleq \left. \frac{\partial f}{\partial \theta_k} \right|_{\theta_k = \hat{\theta}_{k|k}^{(\ell)}}.$$

By plugging in the first order approximation of $f(\theta_k)$ into (3.27b), the expectation term can be written as

$$\mathbb{E}_{\setminus \theta_k} [(a - b\theta_k)^T (sX)^{-1} (a - b\theta_k)],$$

where

$$a \triangleq [f(\hat{\theta}_{k|k}^{(\ell)}) - \nabla f(\hat{\theta}_{k|k}^{(\ell)})\hat{\theta}_{k|k}^{(\ell)}], \quad (3.29)$$

$$b \triangleq -\nabla f(\hat{\theta}_{k|k}^{(\ell)}). \quad (3.30)$$

Through algebraic manipulations, $q_{\theta}^{(\ell+1)}(\theta_k)$ can be expressed as a Gaussian PDF with mean vector $\hat{\theta}_{k|k}^{(\ell+1)}$ and covariance $\Theta_{k|k}^{(\ell+1)}$,

$$q_{\theta}^{(\ell+1)}(\theta_k) = \mathcal{N}(\theta_k; \hat{\theta}_{k|k}^{(\ell+1)}, \Theta_{k|k}^{(\ell+1)}), \quad (3.31)$$

where

$$\hat{\theta}_{k|k}^{(\ell+1)} = \Theta_{k|k}^{(\ell+1)} (\Theta_{k|k-1}^{-1} \hat{\theta}_{k|k-1} + \delta), \quad (3.32a)$$

$$\Theta_{k|k}^{(\ell+1)} = (\Theta_{k|k-1}^{-1} + \Delta)^{-1}, \quad (3.32b)$$

$$\delta = \sum_{j=1}^{m_k} \text{Tr} \left[\overline{s\mathbf{X}_k^{-1}(\mathbf{T}'_{\hat{\theta}_{k|k}^{(\ell)}})^T (\mathbf{z}_k^j - \mathbf{H}\mathbf{x}_k) (\cdot)^T (\mathbf{T}'_{\hat{\theta}_{k|k}^{(\ell)}}) \hat{\theta}_{k|k}^{(\ell)} \right]$$

$$- \text{Tr} \left[s \mathbf{X}_k^{-1} \mathbf{T}_{\hat{\theta}_{k|k}}^{T(\ell)} \overline{(\mathbf{z}_k^j - \mathbf{H} \mathbf{x}_k)(\cdot)^T} (\mathbf{T}'_{\hat{\theta}_{k|k}}) \right], \quad (3.32c)$$

$$\Delta = \sum_{j=1}^{m_k} \text{Tr} \left[s \mathbf{X}_k^{-1} (\mathbf{T}'_{\hat{\theta}_{k|k}})^T \overline{(\mathbf{z}_k^j - \mathbf{H} \mathbf{x}_k)(\cdot)^T} (\mathbf{T}'_{\hat{\theta}_{k|k}}) \right], \quad (3.32d)$$

where $\overline{s \mathbf{X}_k^{-1}} = \mathbb{E}_{q_X^{(\ell)}}[(s \mathbf{X}_k)^{-1}]$, $\overline{(\mathbf{z}_k^j - \mathbf{H} \mathbf{x}_k)(\cdot)^T} = \mathbb{E}_{q_Z^{(\ell)}, q_X^{(\ell)}}[(\mathbf{z}_k^j - \mathbf{H} \mathbf{x}_k)(\cdot)^T]$, and $\mathbf{T}'_{\hat{\theta}_{k|k}} \triangleq \left. \frac{\partial \mathbf{T}_{\theta_k}}{\partial \theta_k} \right|_{\theta_k = \hat{\theta}_{k|k}^{(\ell)}}$.

The derivations of δ and Δ are given in Appendix .1.3.

By using the expressions derived so far, we can set up variational iterations to find the approximate posteriors $q_{\mathbf{x}}$, q_X , q_{θ} , and q_Z . The noise-free measurement set $\mathcal{Z}_k$ can be marginalized out from the joint density, and an approximation for $p(\mathbf{x}_k, \mathbf{X}_k, \theta_k | \mathcal{Y}_{1:k})$ is obtained.

3.3.6 Expectation Calculations

The relevant expectations in the variational iterations can be computed by using the following set of equations:

$$\mathbb{E}_{q_{\mathbf{x}}^{(\ell)}}[\mathbf{x}_k] = \hat{\mathbf{x}}_{k|k}^{(\ell)}, \quad (3.33a)$$

$$\mathbb{E}_{q_Z^{(\ell)}}[\mathbf{z}_k^j] = \hat{\mathbf{z}}_k^{j,(\ell)}, \quad (3.33b)$$

$$\mathbb{E}_{q_X^{(\ell)}}[(s \mathbf{X}_k)^{-1}] = \text{diag} \left(\frac{\alpha^{1,\ell}}{s\beta^{1,\ell}}, \frac{\alpha^{2,\ell}}{s\beta^{2,\ell}}, \dots, \frac{\alpha^{n_y,\ell}}{s\beta^{n_y,\ell}} \right), \quad (3.33c)$$

$$\begin{aligned} \mathbb{E}_{q_Z^{(\ell)}, q_X^{(\ell)}}[(\mathbf{z}_k^j - \mathbf{H} \mathbf{x}_k)(\cdot)^T] &= \mathbf{H} \mathbf{P}_{k|k}^{(\ell)} \mathbf{H}^T + \Sigma_k k^{z,(\ell)} \\ &+ \left(\hat{\mathbf{z}}_k^{j,(\ell)} - \mathbf{H} \hat{\mathbf{x}}_{k|k}^{(\ell)} \right) \left(\hat{\mathbf{z}}_k^{j,(\ell)} - \mathbf{H} \hat{\mathbf{x}}_{k|k}^{(\ell)} \right)^T, \end{aligned} \quad (3.33d)$$

$$\begin{aligned} &\mathbb{E}_{q_{\mathbf{x}}^{(\ell)}, q_{\theta}^{(\ell)}, q_Z^{(\ell)}}[\tilde{\mathbf{z}}_k^j (\tilde{\mathbf{z}}_k^j)^T] \\ &= \mathbb{E}_{q_{\theta}^{(\ell)}} \left[\mathbf{T}_{\theta_k}^T \left(\left(\hat{\mathbf{z}}_k^{j,(\ell)} - \mathbf{H} \hat{\mathbf{x}}_{k|k}^{(\ell)} \right) \left(\hat{\mathbf{z}}_k^{j,(\ell)} - \mathbf{H} \hat{\mathbf{x}}_{k|k}^{(\ell)} \right)^T \right. \right. \\ &\quad \left. \left. + \mathbf{H} \mathbf{P}_{k|k}^{(\ell)} \mathbf{H}^T + \Sigma_k k^{z,(\ell)} \right) \mathbf{T}_{\theta_k} \right], \end{aligned} \quad (3.33e)$$

where the expectation in (3.33e) can be calculated by using the identity $\mathbf{T}_{\theta_k}^T = \mathbf{T}_{-\theta_k}$ and *Lemma 1*.

$$\begin{aligned} \mathbb{E}_{q_X^{(\ell)}}[(s\mathbf{X}_k)] = \\ \text{diag} \left(\frac{s\beta^{1,\ell}}{(\alpha^{1,\ell} - 1)}, \frac{s\beta^{2,\ell}}{(\alpha^{2,\ell} - 1)}, \dots, \frac{s\beta^{n_y,\ell}}{(\alpha^{n_y,\ell} - 1)} \right) \end{aligned} \quad (3.34)$$

The initial conditions for the quantities can be chosen as $\hat{\mathbf{z}}_k^{j,(0)} = \mathbf{y}_k^j$, $\Sigma_k^{z,(0)} = \mathbb{E}_{q_X^{(0)}}[(s\mathbf{X}_k)]$, $\hat{\mathbf{x}}_{k|k}^{(0)} = \hat{\mathbf{x}}_{k|k-1}$, $\mathbf{P}_{k|k}^{(0)} = \mathbf{P}_{k|k-1}$, $\alpha_{k|k}^{(0)} = \alpha_{k|k-1}$ and $\beta_{k|k}^{(0)} = \beta_{k|k-1}$.

3.3.7 Calculation of $\mathbb{E}_{q_X^{(\ell)}, q_\theta^{(\ell)}}[(\mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T)^{-1}]$

This expectation can be calculated exactly, thanks to the factorized distributions.

Lemma 1 *Given*

$$\mathbf{M}^{-1} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix},$$

and $q_\theta^{(\ell)}(\theta_k) = \mathcal{N}(\theta_k, \hat{\theta}_{k|k}^{(\ell)}, \Theta_{k|k}^{(\ell)})$, the entries of the matrix $\mathbb{E}_{q_\theta^{(\ell)}}[(\mathbf{T}_{\theta_k} \mathbf{M} \mathbf{T}_{\theta_k}^T)^{-1}]$ can be computed as:

$$\begin{aligned} \mathbb{E}_{q_\theta^{(\ell)}}[(\mathbf{T}_{\theta_k} \mathbf{M} \mathbf{T}_{\theta_k}^T)^{-1}]_{11} \\ = \begin{bmatrix} m_{11} & m_{22} & -(m_{12} + m_{21}) \end{bmatrix} K \left(\hat{\theta}_{k|k}^{(\ell)}, \Theta_{k|k}^{(\ell)} \right), \end{aligned} \quad (3.35a)$$

$$\begin{aligned} \mathbb{E}_{q_\theta^{(\ell)}}[(\mathbf{T}_{\theta_k} \mathbf{M} \mathbf{T}_{\theta_k}^T)^{-1}]_{12} \\ = \begin{bmatrix} m_{12} & -m_{21} & m_{11} - m_{22} \end{bmatrix} K \left(\hat{\theta}_{k|k}^{(\ell)}, \Theta_{k|k}^{(\ell)} \right), \end{aligned} \quad (3.35b)$$

$$\begin{aligned} \mathbb{E}_{q_\theta^{(\ell)}}[(\mathbf{T}_{\theta_k} \mathbf{M} \mathbf{T}_{\theta_k}^T)^{-1}]_{21} \\ = \begin{bmatrix} m_{21} & -m_{12} & m_{11} - m_{22} \end{bmatrix} K \left(\hat{\theta}_{k|k}^{(\ell)}, \Theta_{k|k}^{(\ell)} \right), \end{aligned} \quad (3.35c)$$

$$\begin{aligned} \mathbb{E}_{q_\theta^{(\ell)}}[(\mathbf{T}_{\theta_k} \mathbf{M} \mathbf{T}_{\theta_k}^T)^{-1}]_{22} \\ = \begin{bmatrix} m_{22} & m_{11} & m_{12} + m_{21} \end{bmatrix} K \left(\hat{\theta}_{k|k}^{(\ell)}, \Theta_{k|k}^{(\ell)} \right), \end{aligned} \quad (3.35d)$$

where

$$K \left(\hat{\theta}_{k|k}^{(\ell)}, \Theta_{k|k}^{(\ell)} \right) \triangleq \begin{bmatrix} 1 + \cos(2\hat{\theta}_{k|k}^{(\ell)}) \exp(-2\Theta_{k|k}^{(\ell)}) \\ 1 - \cos(2\hat{\theta}_{k|k}^{(\ell)}) \exp(-2\Theta_{k|k}^{(\ell)}) \\ \sin(2\hat{\theta}_{k|k}^{(\ell)}) \exp(-2\Theta_{k|k}^{(\ell)}) \end{bmatrix}. \quad (3.35e)$$

The proof is given in Appendix .1.4.

Corollary 1

$$\begin{aligned} & \mathbb{E}_{q_X^{(\ell)}, q_\theta^{(\ell)}} [(s \mathbf{T}_{\theta_k} \mathbf{X}_k \mathbf{T}_{\theta_k}^T)^{-1}] \\ &= (1 - \exp(-2\Theta_{k|k}^{(\ell)})) \frac{\text{Tr}(\mathbb{E}_{q_X^{(\ell)}} [(s \mathbf{X}_k)^{-1}])}{2} \mathbb{I}_2 \\ &+ \exp(-2\Theta_{k|k}^{(\ell)}) \left(\mathbf{T}_{\hat{\theta}_{k|k}^{(\ell)}} \mathbb{E}_{q_X^{(\ell)}} [(s \mathbf{X}_k)^{-1}] \mathbf{T}_{\hat{\theta}_{k|k}^{(\ell)}}^T \right), \end{aligned} \quad (3.36)$$

where $\mathbb{I}_2$ is 2x2 identity matrix. This expression is obtained from Lemma 1 by exploiting the fact that the matrix $\mathbf{X}_k$ is diagonal by definition.

A summary of the resulting iterative measurement update procedure is given in Algorithm 3.1.

Algorithm 3.1 Variational Measurement Update

Given $\hat{\mathbf{x}}_{k|k-1}$, $\mathbf{P}_{k|k-1}$, $\{\alpha_{k|k-1}^i, \beta_{k|k-1}^i\}_{i=1}^{n_y}$, $\hat{\theta}_{k|k-1}$, $\Theta_{k|k-1}$ and $\mathbf{y}_k$; calculate $\hat{\mathbf{x}}_{k|k}$, $\mathbf{P}_{k|k}$, $\{\alpha_{k|k}^i, \beta_{k|k}^i\}_{i=1}^{n_y}$, $\hat{\theta}_{k|k}$, $\Theta_{k|k}$ as follows.

Initialization

$$\begin{aligned} \hat{\mathbf{x}}_{k|k}^{(0)} &\leftarrow \hat{\mathbf{x}}_{k|k-1}, & \mathbf{P}_{k|k}^{(0)} &\leftarrow \mathbf{P}_{k|k-1}, \\ \hat{\theta}_{k|k}^{(0)} &\leftarrow \hat{\theta}_{k|k-1}, & \Theta_{k|k}^{(0)} &\leftarrow \Theta_{k|k-1}, \\ \alpha_{k|k}^{i,(0)} &\leftarrow \alpha_{k|k-1}^i, & \beta_{k|k}^{i,(0)} &\leftarrow \beta_{k|k-1}^i \text{ for } i = 1, \dots, n_y, \\ \mathbf{z}_{k|k}^{j,(0)} &\leftarrow \mathbf{y}_k^j \text{ for } j = 1, \dots, m_k, \\ \Sigma_k k^{z,(0)} &\leftarrow \mathbb{E}_{q_X^{(0)}} [(s \mathbf{X}_k)] \text{ using (3.34)} \end{aligned}$$

Iterations:

for $\ell = 0, \dots, \ell_{max} - 1$ **do**

 Calculate the expectations in (3.33), and (3.36)

 Update $\hat{\mathbf{x}}_{k|k}^{(\ell+1)}$, and $\mathbf{P}_{k|k}^{(\ell+1)}$ using (3.17)

 Update $\hat{\theta}_{k|k}^{(\ell+1)}$, and $\Theta_{k|k}^{(\ell+1)}$ using (3.32)

 Update $\alpha_{k|k}^{i,(\ell+1)}$, and $\beta_{k|k}^{i,(\ell+1)}$ using (3.21) for $i = 1, \dots, n_y$

 Update $\hat{\mathbf{z}}_{k|k}^{j,(\ell+1)}$, and $\Sigma_k k^{z,(\ell+1)}$ using (3.25) for $j = 1, \dots, m_k$

end for

Set final estimates:

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k}^{(\ell_{\max})}, & \mathbf{P}_{k|k} &= \mathbf{P}_{k|k}^{(\ell_{\max})}, \\
\hat{\theta}_{k|k} &= \hat{\theta}_{k|k}^{(\ell_{\max})}, & \Theta_{k|k} &= \Theta_{k|k}^{(\ell_{\max})}, \\
\alpha_{k|k}^i &= \alpha_{k|k}^{i,(\ell_{\max})}, & \beta_{k|k}^i &= \beta_{k|k}^{i,(\ell_{\max})} \text{ for } i = 1, \dots, n_y
\end{aligned}$$

3.4 Time Update

Once the measurement update is performed, the sufficient statistics of the posterior density must be propagated in time in accordance with the target dynamics. An optimal time update step requires the solution to the following Chapman-Kolmogorov equation

$$\begin{aligned}
p(\mathbf{x}_k^a, \mathbf{X}_k | \mathcal{Y}_{1:k-1}) &= \int p(\mathbf{x}_k^a, \mathbf{X}_k | \mathbf{x}_{k-1}^a, X_{k-1}) \\
&\quad p(\mathbf{x}_{k-1}^a, X_{k-1} | \mathcal{Y}_{1:k-1}) d\mathbf{x}_{k-1}^a dX_{k-1},
\end{aligned} \tag{3.37}$$

where $\mathbf{x}_k^a \triangleq [\mathbf{x}_k^T \theta_k]^T$. Unfortunately, it is not possible to obtain an exact compact form analytical expression for most extended target tracking models. Therefore various independence conditions are implied to perform time updates in the literature [18, 25, 42, 44]. For a detailed analysis of possible time update approaches, interested readers can refer to [25] and the references therein.

In the random matrix framework, it is possible to assume that the dynamical models of the kinematic state and the extent state are independent [18],

$$p(\mathbf{x}_k, \mathbf{X}_k | \mathbf{x}_{k-1}, X_{k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{X}_k | X_{k-1}). \tag{3.38}$$

Consequently, the time update of the kinematic state and the extent state can be decoupled for factorised posteriors. The time update of the kinematic state follows the Kalman filter prediction equations if the underlying dynamics are linear. Consider the following state space model which describes the dynamics of the augmented state vector $\mathbf{x}_k^a$,

$$\mathbf{x}_k^a = \mathbf{F} \mathbf{x}_{k-1}^a + u_k, \quad u_k \sim \mathcal{N}(0, \mathbf{Q}). \tag{3.39}$$

The prediction density $\mathcal{N}(\mathbf{x}_{k|k-1}^a; \hat{\mathbf{x}}_{k|k-1}^a, P_{k|k-1}^a)$ is obtained by updating the sufficient statistics (mean and covariance) of the Gaussian components in accordance with

the system dynamics

$$\hat{\mathbf{x}}_{k|k-1}^a = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1}^a, \quad (3.40a)$$

$$\mathbf{P}_{k|k-1}^a = \mathbf{F}\mathbf{P}_{k-1|k-1}^a\mathbf{F}^T + \mathbf{Q}. \quad (3.40b)$$

where $\mathbf{P}_k^a \triangleq \text{blkdiag}(\mathbf{P}_k, \Theta_k)$.

In most tracking applications, the exact dynamics of the extent state is unknown. Even in the case where the dynamic equations of the extent states are available, the transition density induced by the known dynamics may not lead to a prediction update that results in the same family of probability distributions using (3.37). If the dynamics of the extent state is slowly varying but unknown, it is possible to obtain the maximum entropy prediction density of the extent states by utilizing a forgetting factor [62, Theorem 1]. In that case, the sufficient statistics of the inverse Gamma distribution is updated as

$$\alpha_{k|k-1}^i = \gamma_k \alpha_{k-1|k-1}^i, \quad (3.41a)$$

$$\beta_{k|k-1}^i = \gamma_k \beta_{k-1|k-1}^i, \quad \text{for } i = 1, \dots, n_y \quad (3.41b)$$

where γ is the forgetting factor. We prefer to use the maximum entropy prediction density in the time update. However, it is possible to perform alternative time updates within the proposed framework.

3.5 A Closer Look to a Single Measurement Update

In this section, we investigate the proposed measurement update, here and after denoted as VB, in more detail and illustrate its capabilities in comparison with a state-of-the-art extended Kalman filter (EKF) algorithm [90]. For this purpose, we initiate the prior mean and covariance of both approaches the same; and we compare the posterior distribution of the extent states. Consider the example given in Figure 3.2, where the prior mean of the target's location is $[-20 \ -20]^T$. The measurements are shown with blue stars, and the posterior means of the VB and EKF updates are shown with the solid green and red lines, respectively. The median of the true posterior, which is computed by using 1 million Monte Carlo samples, is shown with the solid black

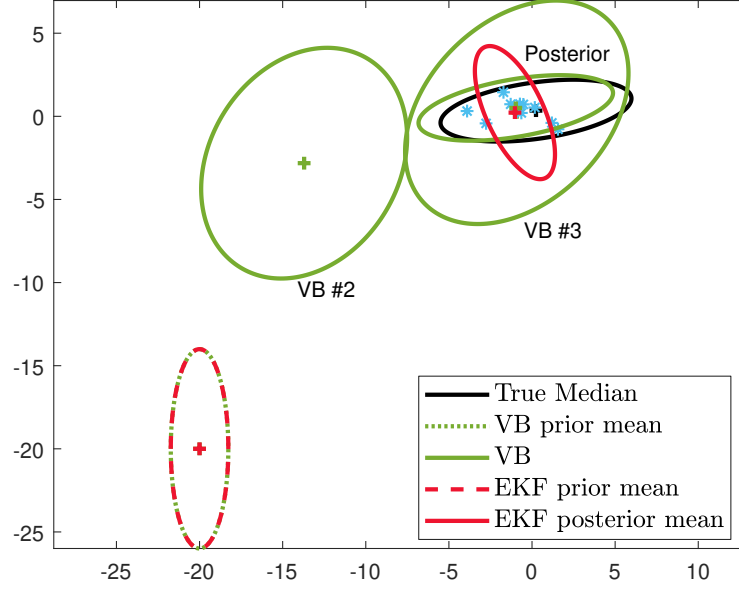


Figure 3.2: A single measurement update for VB and the EKF approach. The prior and posterior mean shape estimates are represented by green dotted and solid lines for VB, respectively. The red dashed line indicates the prior mean shape estimate while the red solid line depicts the posterior mean estimate for EKF approach. The VB $\#i$ denotes the i^{th} variational iteration shape estimate mean of the VB algorithm.

line. The mean of the extent and kinematic state distributions at the end of each VB iteration is denoted by VB $\#i$ where $\#i$ stands for the i^{th} variational iteration. A total of 10 iterations are performed within the variational update. As shown in Figure 3.2, the posterior found by the VB algorithm is closer to the true posterior than the posterior computed by the EKF, thanks to the iterative nature of the VB updates. Unlike EKF, the VB algorithm performs multiple iterations in a single update and performs multiple linearizations during the iterations by taking all available measurements into account. The ability to compute the posterior iteratively is the key concept to explain the superior performance of VB in the experiments given in Section 3.6.

Lastly, we compare the average computation time of the algorithms. The simulations for the illustrative example are run in Matlab(R) R2019b on a standard laptop with an Intel(R) Core(TM) i7-6700HQ 2.60 GHz platform with 16 GB of RAM. We compare naive implementations of the algorithms without exploiting any code optimization methods. A single measurement update (with 10 iterations) and a single variational

iteration for VB takes 2.3×10^{-3} sec and 2.1×10^{-4} sec, respectively. On the other hand, it takes 8.6×10^{-4} sec to perform a measurement update for the EKF. The relevant parameters of the illustrative example are given in the Appendix-.1.5.

3.6 Experimental Results

In this section, we evaluate the performance of the proposed method and compare it with relevant elliptical object tracking algorithms in the literature. The comparison is performed through both simulations and real data experiments. The alternative models are selected as the state-of-the-art EKF approach that is capable of tracking the orientation of elliptical objects [90] and the widely used RM based ETT model [18]. In the sequel, we denote these algorithms as Algorithm-1 and Algorithm-2, respectively. The simulation results are presented in Section 3.6.1, and the results of the real-data experiment are given in Section 3.6.4.

3.6.1 Simulations

In the simulations, we use the Gaussian Wasserstein (GW) distance [19], [91] and root-mean-square-error (RMSE) for performance evaluation and comparison,

$$\begin{aligned} \text{GW}(\mathbf{m}_a, \mathbf{X}_a, \mathbf{m}_b, \mathbf{X}_b)^2 & \\ \triangleq & \underbrace{\|\mathbf{m}_a - \mathbf{m}_b\|_2^2}_{1^{\text{st}} \text{ Term}} + \underbrace{\text{Tr}[\mathbf{X}_a + \mathbf{X}_b - 2(\mathbf{X}_a^{\frac{1}{2}} \mathbf{X}_b \mathbf{X}_a^{\frac{1}{2}})^{\frac{1}{2}}]}_{2^{\text{nd}} \text{ Term}}. \end{aligned} \quad (3.42)$$

Here, $\mathbf{m}_a$, $\mathbf{m}_b$ and $\mathbf{X}_a$, $\mathbf{X}_b$ stand for two different center locations and elliptic extent matrices, respectively. The first term in (3.42) corresponds to the error in the estimation of the object's center, and the second term corresponds to the error in extent estimation. We report both terms in (3.42) in addition to the overall GW distance to provide insight into the estimation performance of the algorithms in detail. Furthermore, we compare the RMSE of the orientation estimations which is defined by

$$\text{RMSE}(\theta_{\text{true}}, \theta) = \sqrt{\frac{1}{N} \sum_{k=1}^N (\theta_{k,\text{true}} - \theta_k)^2}, \quad (3.43)$$

where N denotes the number of time steps in a single run.

3.6.2 Constant Velocity Model

In the first experiment, a dynamic object is simulated, which moves according to the nearly constant velocity model defined by the following parameters.

$$\bar{\mathbf{F}} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes \mathbb{I}_2, \quad \mathbf{F} = \text{blkdiag}(\bar{\mathbf{F}}, 1), \quad (3.44a)$$

$$\mathbf{P}_0 = \mathbb{I}_5, \quad \hat{\mathbf{x}}_0 = \begin{bmatrix} 0 & 0 & 50 & 0 & 0 \end{bmatrix}^T, \quad (3.44b)$$

$$\mathbf{X}_{true} = \begin{bmatrix} 50 & 0 \\ 0 & 600 \end{bmatrix}, \quad \bar{\mathbf{Q}} = \sigma^2 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \otimes \mathbb{I}_2, \quad (3.44c)$$

$$\mathbf{Q} = \text{blkdiag}(\bar{\mathbf{Q}}, \sigma_\theta), \quad \mathbf{R} = 5 \times \mathbb{I}_2, \quad (3.44d)$$

where $T = 0.1$, $\sigma = 1$, and $\sigma_\theta = 0.01$. In this simulation, the parameters of the motion model are fully provided to the tracking algorithms so that the error due to model-mismatch does not affect the estimation performance. Throughout the trajectory, the object generates an average of 10 measurements per scan. We investigate two different cases separately; in the first case, the measurements follow a Gaussian distribution, and in the second case, they follow a uniform distribution. All simulation experiments were performed 100 times with different realizations of the process noise, measurement noise, and measurement origin at each simulation. The presented numbers are the average of these 100 Monte Carlo (MC) runs. The algorithm specific initial shape variables for VB are set to $\alpha_0^{1,2} = [2 \ 2]^T$ and $\beta_0^{1,2} = [100 \ 100]^T$. The number of variational iterations is 10.

The shape variables are initialized for Algorithm-2 as $v_0 = 4$ and $\mathbf{V}_0 = 100 \times \mathbb{I}_2$. The forgetting factor is set to $\gamma = 0.99$ for both VB and Algorithm-2. To be consistent with [90], we use the same notations for the parameters of Algorithm-1. The prior mean and covariance matrix of the shape variables of Algorithm-1 are selected to be $\hat{\mathbf{p}}_0 = [0 \ 10 \ 10]^T$ and $\mathbf{C}_0^p = \text{diag}([1, \ 20, \ 20])$. The vector $\hat{\mathbf{p}}_0$ consists of $[\theta, \ l_1, \ l_2]$ where, θ , l_1 , and l_2 are the orientation and the semi-axis lengths, respec-

Table 3.2: The GW distance values for Gaussian measurements.

	GW Distance 1 st Term [m ²]	GW Distance 2 nd Term [m ²]	GW Distance [m]	
			Mean	Std.
Algorithm-1	4.78	8.54	3.18	0.24
Algorithm-2	5.22	56.63	6.86	1.68
VB	4.49	5.27	2.85	0.23

Table 3.3: The GW distance values for uniformly distributed measurements.

	GW Distance 1 st Term [m ²]	GW Distance 2 nd Term [m ²]	GW Distance [m]	
			Mean	Std.
Algorithm-1	1.93	6.02	2.49	0.55
Algorithm-2	2.18	58.54	6.69	1.65
VB	1.92	4.54	2.28	0.41

tively. The process noise covariance matrix for the shape variables for Algorithm-1 is $\mathbf{C}_p^w = \text{diag}([10^{-2}, 0.1, 0.1])$. The kinematic state transition matrix for Algorithm-1 is set to $\mathbf{A}_r = \mathbf{F}(1 : 4, 1 : 4)$. The initial mean of the kinematic state vector is the same as VB, $\hat{\mathbf{r}}_0 = \hat{\mathbf{x}}_0(1 : 4)$. The state transition matrix for the shape variables is $\mathbf{A}_p = \mathbb{I}_3$. The initial values of the shape and kinematic variables are selected to make the prior means of the algorithms the same. The algorithmic specific parameters are hand-tuned to obtain the best performance of each algorithm. We report the average GW distance and the orientation RMSE for Gaussian and uniformly distributed measurements in Table 3.2 and Table 3.3, respectively. The proposed algorithm performs better in terms of estimating the extent of the target and outperforms the other algorithms in terms of GW distance. Additionally, VB shows a better performance in estimating the orientation of the target.

Table 3.4: The heading angle RMSE values for Gaussian and uniform measurements.

	Gaussian Measurements		Uniform Measurements	
	Heading Angle RMSE [°]		Heading Angle RMSE [°]	
	Mean	Std.	Mean	Std.
Algorithm-1	4.46	0.87	4.93	0.45
Algorithm-2	59.98	34.38	60.15	34.00
VB	3.93	0.41	4.00	0.43

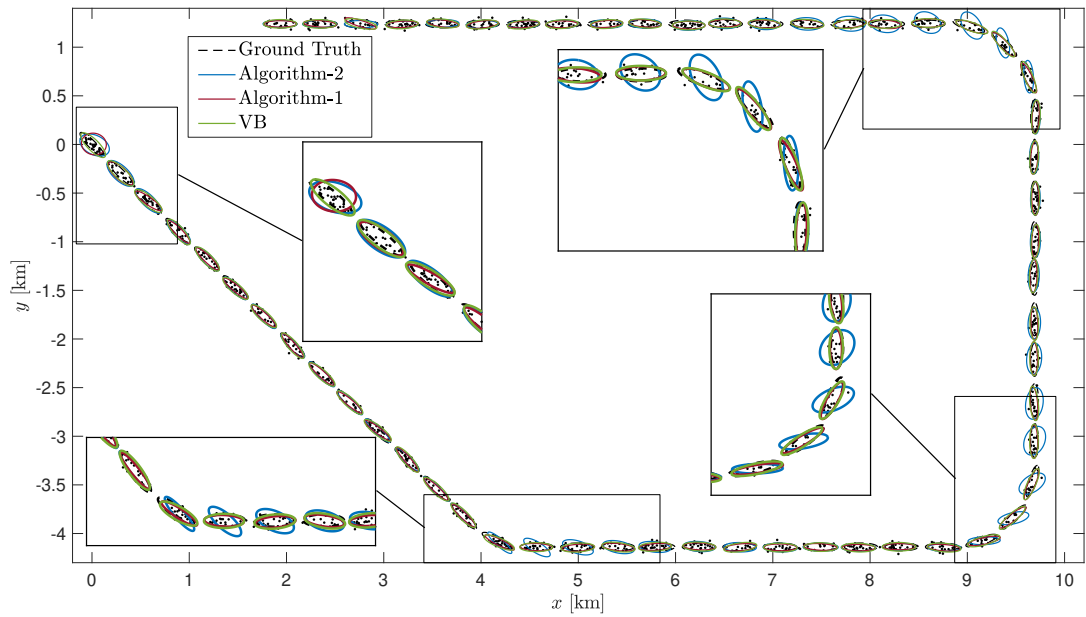


Figure 3.3: An example MC run of the scenario in Section 3.6.3

3.6.3 Experimental Trajectory

This experiment involves the scenario studied in [18,44,89,90]. In this simulation, the trajectory composed of one 45° and two 90° turns pieced together with straight paths. The object of interest has unknown but fixed semi-axes lengths, and its orientation varies in time. The object starts its motion from the origin with a speed of 50 km/h, which is fixed throughout the trajectory. The measurements are generated from a uniform distribution, and the number of the measurements is drawn from a Poisson distribution with an average of 20 measurements per scan. In addition to simulations performed in [90], we will examine the performance of the algorithms with Gaussian distributed measurements. As in [90], the prior mean and covariance matrix of the shape variables are selected to be $\hat{\mathbf{p}}_0 = [\pi, 200, 90]^T$ and $\mathbf{C}_0^p = \text{diag}([1, 70^2, 70^2])$. The process noise covariance matrix for the shape variables and kinematics are $\mathbf{C}_p^w = \text{diag}([0.1, 1, 1])$ and $\mathbf{C}_r^w = \text{diag}([100, 100, 1, 1])$, respectively. The measurement noise covariance matrix is, $\mathbf{R} = \text{diag}([400, 400])$. In order to have a fair comparison, the prior mean values of the kinematic and shape variables for VB and Algorithm-2 are chosen to be the same as those of Algorithm-1. The shape variables for VB are selected to be $\alpha_0^{1,2} = [5 \ 5]$ and $\beta_0^{1,2} = [400^2 \ 180^2]$. The degrees of freedom is $\nu_0 = 7$ for Algorithm-2. The scale matrix is initialized as $\mathbf{V}_0 = \text{diag}([400^2, 180^2])$. The initial mean of the kinematic state is set to $\hat{\mathbf{x}}_0 = [100 \ 100 \ 5 \ -8 \ \pi]^T$ for VB. The number of the variational iterations is 10. The initial mean of the kinematic state for the Algorithm-1 and Algorithm-2 is selected to be $\hat{\mathbf{r}}_0 = \hat{\mathbf{x}}_0(1 : 4)$.

We conducted 100 MC runs for each measurement distribution type. The GW distance and the orientation RMSE are presented in Table 3.5 and Table 3.6. An example MC run is depicted in Fig 3.3. Algorithm-2 could not perform well during the turns because the method does not treat the orientation as a separate random variable and compensates the changes in the orientation by updating the extent estimate. However, VB and Algorithm-1 are able to overcome this problem. The results show that the proposed approach, VB, provides better orientation, center, and extent estimates.

Table 3.5: The GW distance and heading angle RMSE values of the scenario in Section 3.6.3 when the measurements are uniformly distributed.

	GW Dist. 1 st Term [m ²]	GW Dist. 2 nd Term [m ²]	GW Dist. [m]		Heading Angle RMSE [°]	
			Mean	Std.	Mean	Std.
Alg.-1	281.38	280.54	20.84	0.94	3.89	0.26
Alg.-2	284.05	1436.05	32.94	082	82.61	5.21
VB	270.74	203.01	19.83	0.89	3.37	0.21

Table 3.6: The GW distance and heading angle RMSE values of the scenario in Section 3.6.3 when the measurements are generated from a Gaussian distribution.

	GW Dist. 1 st Term [m ²]	GW Dist. 2 nd Term [m ²]	GW Dist. [m]		Heading Angle RMSE [°]	
			Mean	Std.	Mean	Std.
Alg.-1	884.75	244.15	29.17	1.43	3.57	0.35
Alg.-2	826.32	1167.80	37.69	1.29	82.66	5.08
VB	822.15	145.31	27.36	1.39	2.94	0.20

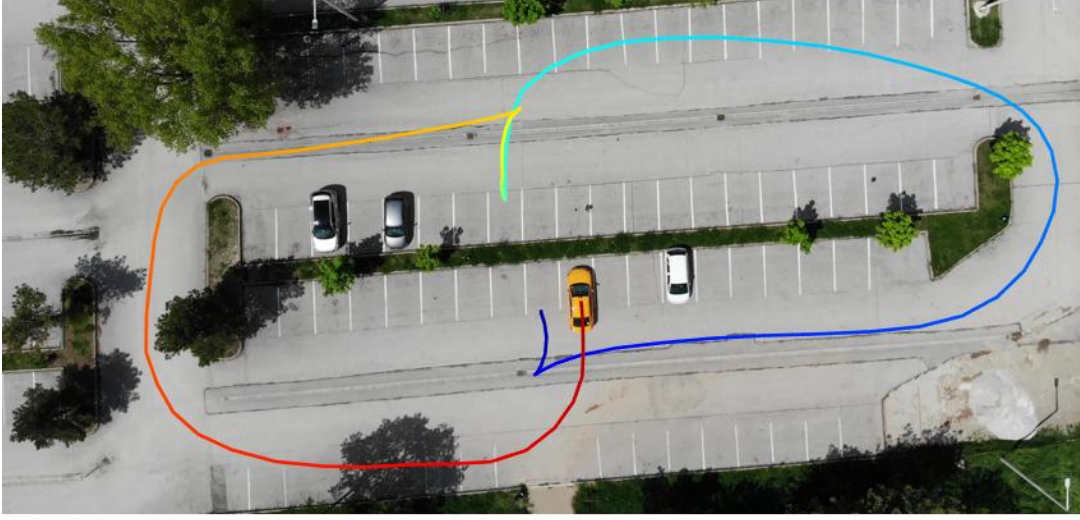


Figure 3.4: The outline of the movement of the vehicle during the time-lapse. The vehicle starts from the dark blue colored parking spot; and follows the colored path until the red colored parking spot. In the figure, the last frame is shown.

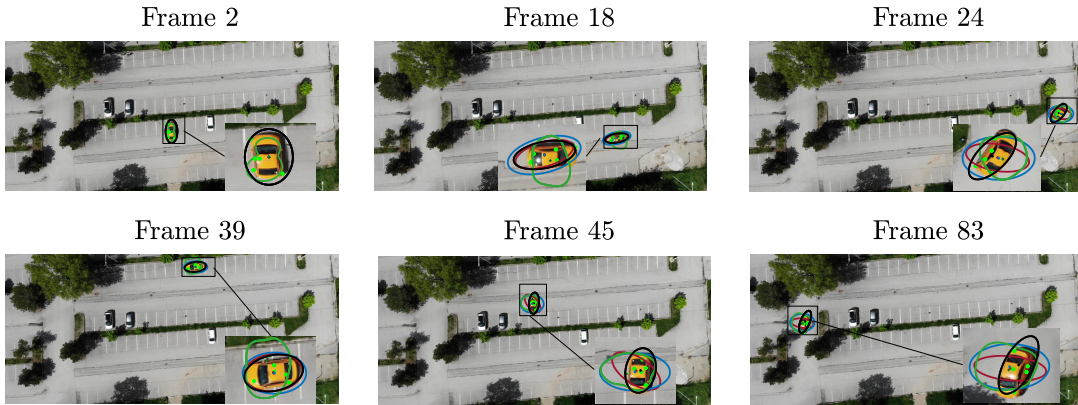


Figure 3.5: A representative MC run of the real data experiment. The extent estimates of VB, Algorithm-1, Algorithm-2, and GP-ETT are shown in black, red, blue, and green lines, respectively. The measurements are represented with green dots.

3.6.4 Real Data Experiment

In this section, the algorithms' capabilities are illustrated with real data. In addition to elliptical models, we compare the performance of another well-known ETT algorithm, namely the Gaussian process based extended target tracking (GP-ETT) algorithm [82] to demonstrate the performance of the methods that do not rely on elliptical extent assumption in the scenario.

The test data is collected in an urban area of Ankara. The test scenario involves a commercial vehicle moving in a parking lot while a steady aerial camera captures images of the surveillance region every second, i.e., $T = 1s$. In the scenario, a long sampling time is intentionally chosen to minimize the computational power consumption, thereby prolonging the air-time of the aerial camera in possible real-time applications. The outline of the vehicle's trajectory is shown in Figure 3.4. The colored-line indicates the trajectory followed by the mid-point of the vehicle. The scenario starts while the vehicle is parked in the parking area, indicated by the dark blue color. The vehicle leaves the parking area and follows the path shown in blue until it is parked in the parking area, which is indicated by the green color. Then the vehicle performs a similar motion from the green-colored parking spot following the path to the red-colored parking spot.

Throughout the scenario, the captured images are processed for measurement extraction. Various feature extraction algorithms can be used to obtain measurements from the vehicle such as Harris corner detection [31], Scale Invariant Feature Transform (SIFT) [50], Speeded Up Robust Features (SURF) [7] or similar. In order to demonstrate that the algorithm can work with a wide range of feature extraction algorithms, we present a more general case, where the measurements are uniformly sampled from the vehicle's visible surface. The results obtained by using the features extracted by the Harris corner detector are also consistent with the results presented here.

As part of the image processing step, a segmentation is performed in every frame in the HSV color band to separate the yellow vehicle from the background. Following that, a median filter is used to reduce the number of clutters. Finally, the pixels

that belong to the vehicle are sampled uniformly to obtain the measurements. The initial position of the vehicle is extracted from the first frame. The initial velocity, on the other hand, is assumed to be unknown and assumed to be zero. Hence, the initial mean of the kinematic state vector is selected to be $\hat{\mathbf{x}}_0 = [450 \ 245 \ 0 \ 0 \ \frac{\pi}{2}]^T$. The initial parameters of the algorithms are selected to match the prior means of the corresponding distributions. For this purpose, the initial shape parameters for VB is selected to be $\alpha_0^{1,2} = [2 \ 2]^T$ and $\beta_0^{1,2} = [250 \ 1000]^T$. The prior mean and covariance matrix of the shape variables for Algorithm-1 is set to $\hat{\mathbf{p}}_0 = [0 \ 250^{0.5} \ 1000^{0.5}]^T$ and $\mathbf{C}_0^p = \text{diag}([1, 100, 100])$, respectively. The degrees of freedom value and the initial scale matrix is set to $v_0 = 4$ and $\mathbf{V}_0 = \text{diag}([250, 1000])$ for Algorithm-2, respectively. The process noise covariance matrix $\mathbf{Q}$ is similar to the previous simulations, however σ is taken as 4, and σ_θ is 0.1 for VB. The number of variational iterations is 10. The process noise covariance matrix for the shape variables for Algorithm-1 is set to $\mathbf{C}_p^w = \text{diag}([0.1, 10^{-3}, 10^{-3}])$. Finally, the measurement noise covariance matrix is taken as $\mathbf{R} = \text{diag}([1, 1])$. The parameters were optimized manually to obtain the best performances of the algorithms.

The extent estimates corresponding to the frames $\{2, 18, 24, 39, 45, 83\}$ are given in Figure 3.5. These snapshots were chosen for the sake of a clearer illustration of the differences between the algorithms' performances, starting from the initial frames. At the beginning of the scenario, the vehicle stays immobile, and the algorithms are able to estimate the vehicle's extent satisfactorily (see: Frame 2).

Before the vehicle starts its movement, the extent estimate of the GP-ETT is more accurate and closer to the true extent of the target. On the other hand, the random matrix approaches are significantly advantageous throughout this scenario because they can fully exploit the prior information that the target's extent is close to an elliptical shape. GP-ETT algorithm aims at estimating the contours of objects having arbitrary shapes and its performance degrades when the measurements are originating from the surface of objects and the number of the surface measurements is low. The algorithm is essentially trying to solve a harder problem because it has more degrees of freedom to represent the unknown extent and a greater uncertainty to resolve compared to the

ellipsoidal target tracking methods.

When the vehicle is moving in a straight path, such as in Frame 18 and Frame 39, the performance of all random matrix based algorithms are satisfactory. However, when the vehicle performs a maneuver, as in Frame 24, Frame 45 and Frame 83, VB shows superior performance in estimating the orientation of the vehicle. During the maneuvers, Algorithm-2 cannot estimate the extent accurately because it does not treat the heading angle as a separate random variable, and it tries to adapt to the changes in the orientation by updating the extent states. Algorithm-1 also struggles to find the correct orientation of the vehicle. However, VB can provide accurate estimates of the extent thanks to its iterative updates. Note that VB and Algorithm-1 use the same process noise variance for the orientation. Since the vehicle is stable in the first couple of frames and the algorithms are able to estimate the extent accurately, increasing the variance values of the shape variables for Algorithm-1 does not improve the performance of estimating the extent further. Additionally, if the variance values are increased too much, the extent estimates of Algorithm-1 tend to collapse to zero. We encountered a similar problem while tuning Algorithm-1 in the simulation scenarios. We report one example of such behavior in a single measurement update in Appendix- .1.6 for interested readers.

3.7 Conclusion and Discussion

ETT involves tracking objects that generate multiple measurements per scan. In most ETT applications, the orientation of the extended targets changes in time. In standard RM based ETT methods, this phenomenon is addressed by a forgetting factor, which aims at forgetting the accumulated information. In this study, we proposed a novel approach for extended target tracking that is capable of simultaneously estimating the kinematic, extent, and orientation states of an extended target. We use the variational Bayes technique for inference and define appropriate priors for the unknown state variables that can accurately model the changes in the extended targets' orientation. The performance and capabilities of the algorithm are demonstrated through

simulations and real data experiments. Experimental results on simulations and real data demonstrate that the proposed method significantly improves the tracking performance, as well as the accuracy in estimating the orientation and the shape of the object compared to the state-of-the-art methods.

It is also worth mentioning that variational Bayes approaches resort to factorized distributions, which lose the correlation structure in the posterior density. An algorithm that does not neglect correlation terms may provide better estimation performance than such variational methods. In our experience (as illustrated in benchmark scenarios and real data experiments in Sections 3.5-3.6), the iterative optimization structure provided by the variational inference framework outperforms the alternative solutions by overcoming the disadvantages associated with the factorized approximation. However, improved performance can be achieved by further exploiting the correlation structure in the true posterior.

The model and the technique we use might also be applicable to estimation problems other than extended target tracking applications, which involve dynamic elliptical representations or unknown covariance matrices with similarly structured uncertainty. These problems may include, but are not limited to, obtaining elliptical bounds in power systems [14, 67], estimating ellipsoid sets containing target states over sensor networks [16] or spectrum representation in speech processing [12, 61].

CHAPTER 4

MULTI-ELLIPSOIDAL EXTENDED TARGET TRACKING WITH VARIATIONAL BAYES INFERENCE

4.1 Introduction

Recent advances in autonomous vehicles, robotics, and intelligent systems require not only estimating the position of an object but also recognizing its shape. This requirement is usually fulfilled with short-range sensor systems, where it is possible to collect multiple instantaneous measurements from a single target. In contrast to traditional point target tracking methods, one can extract more information from the measurements, such as the shape, size, or orientation of the target. The special algorithms that are capable of estimating these unknowns together with the target's kinematic state are called Extended Target/Object Tracking (ETT/EOT) algorithms.

A primitive approach to ETT is to represent the target's extent as a simple shape and estimate the relevant parameters. These simple shapes can be a line [23], a circle [6] or a rectangle [24]. More complex shapes can be defined using *random hyper-surface models* which assume that the measurements are generated from an unknown random surface [4, 5, 90]. More recently, Gaussian Process (GP) based models [43, 82] were proposed to estimate the extent of targets with unknown shapes. GP based ETT algorithms define the target's contour as an unknown radial function with a GP prior. An advantage of these models is that the estimated contours are descriptive, i.e., the contour representations can be utilized in further purposes such as classification of targets [76, 77].

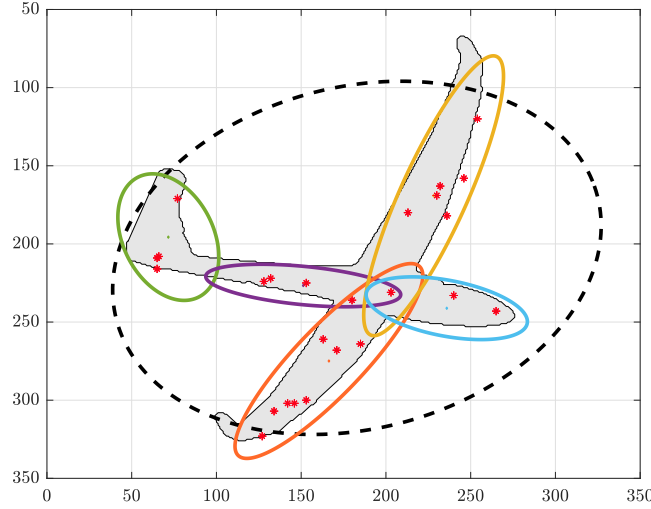


Figure 4.1: An extended target representation with multiple ellipses (solid lines). Classical RM models represent the same target extent as a single ellipse (dashed black line). The measurements are shown with red stars.

One of the most common approaches in the ETT literature is the random matrix model (RM) [18, 27, 34, 39, 42, 45, 60, 78, 88–90], which was pioneered by Koch [42]. In RM based methods, the extent is approximated by an ellipse which is represented by a symmetric positive definite (SPD) unknown matrix. The inference in [42] neglects the measurement noise covariance matrix in order to meet the conjugacy requirement. This approach was later improved in [18], where the measurement noise covariance is incorporated into the updates. More recent studies consider the orientation angle of the target together with the RM model [78, 88–90].

A single ellipsoidal representation of the target can provide only lumped information about the shape of the target, which may result in over-simplified representations. To remedy this problem, more recent works on random matrices focus on multi-ellipsoidal (ME) models [27, 34, 39, 45], where the target extent is represented with more than one ellipse. Each ellipse is called a sub-object, and as the number of sub-objects increases, a finer representation of the target’s extent can be obtained. A representative example of using multiple ellipses for target extent is given in Fig. 4.1.

ME models in the literature use different inference methods for estimating a mixture of Gaussian inverse Wishart (GIW) distributions which represent the kinematic state

together with the extent. In [39], a particle filter is used for inference which becomes computationally heavy when the number of particles is increased. The ME model proposed in [45] does not contain a unified kinematic model for different sub-objects. In [27], the kinematics of the sub-objects are unified however, the method requires computationally heavy partitioning algorithms to associate the measurements with the sub-objects. Furthermore, mixture reduction algorithms are required to manage the number of components resulting from different association hypotheses. In [34], an approach that can handle varying numbers of sub-objects in time is proposed.

In this chapter, we present a ME-ETT approach utilizing variational Bayes inference for solving the measurement association problem and obtaining an approximate distribution for the intractable posterior. Our approach does not require any clustering [45], partitioning [27], mixture reduction [27] and merging methods [45]. The resulting algorithm has low computational complexity and outperforms the state-of-the-art methods in terms of accuracy. It can be employed in scenarios where the users require a more detailed representation of the targets, such as tracking of aerial vehicles, see Fig. 4.1. Another well-suited reason for use would be tracking a group of targets, such as a flock of birds or a group of people, whose individuals share the same kinematic behaviour so that the position of every member of the group can be estimated well by the proposed algorithm even when a member of the group is obstructed from the vision of the sensor. The proofs of concepts for the aforementioned use cases are illustrated in the coming sections.

The rest of this part is organized as follows. First, the problem formulation is given in Section 4.2. Then, the variational inference for the measurement update is explained in Section 4.3. Section 4.4 presents the time update. In Section 4.5 the simulation results will be shown and discussed. Finally, we will conclude the article in Section 4.7.

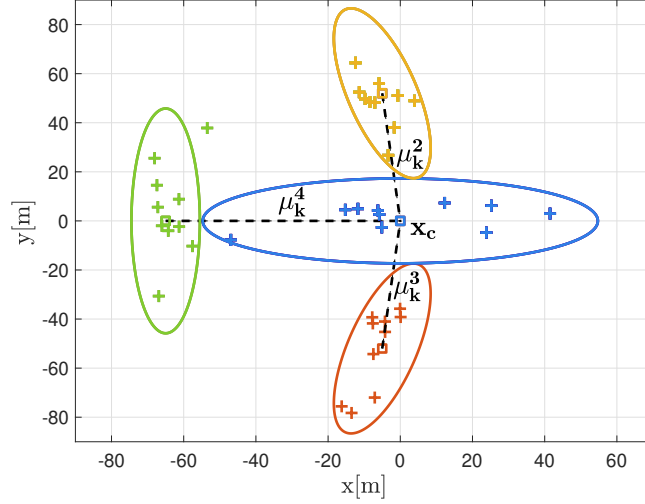


Figure 4.2: Illustration of the extent estimation for $L = 4$ where $+$ sign denotes the measurements.

4.2 Problem Formulation

We consider the following extended target model, first proposed in [39], composed of $L \geq 1$ sub-objects. At any time k , the extent state of the ℓ th sub-object is represented by the SPD matrix $\mathbf{X}_k^\ell \in \mathbb{R}^{n_y \times n_y}$, $\ell = 1, \dots, L$. We define the kinematic state of the extended object as

$$\mathbf{x}_k \triangleq \begin{bmatrix} (\mathbf{x}_k^c)^T & \tilde{\mathbf{x}}_k^T & (\boldsymbol{\mu}_k^2)^T & \dots & (\boldsymbol{\mu}_k^L)^T \end{bmatrix}^T \quad (4.1)$$

where $\mathbf{x}_k^c \in \mathbb{R}^{n_y}$ denotes a reference point on the target and $\{\boldsymbol{\mu}_k^\ell \in \mathbb{R}^{n_y}\}_{\ell=1}^L$ denote the displacement vectors of the sub-object centers from the reference point $\mathbf{x}_k^c$. We assume that $\boldsymbol{\mu}_k^1 = \mathbf{0}$, i.e., the reference point of the extended target is at the center of the first sub-object without loss of generality. The vector $\tilde{\mathbf{x}}_k$ contains all non-positional kinematic information, such as velocity and acceleration. An illustration of this kinematic state vector for $L = 4$ is shown in Fig. 4.2. The kinematic state vector $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is assumed to evolve with the following linear Gaussian dynamics

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k, \quad (4.2)$$

Table 4.1: Notations

-
- Set of real matrices of size $m \times n$ is shown with $\mathbb{R}^{m \times n}$.
 - n_y and n_x are the dimension of the measurement vector and the dimension of the kinematic state vector, respectively.
 - $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the the multivariate Gaussian distribution with mean vector $\boldsymbol{\mu} \in \mathbb{R}^{n_x}$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{S}_{++}^{n_x}$. $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is used for the value of this distribution at $\mathbf{x} \in \mathbb{R}^{n_x}$.
 - $\mathcal{IW}(v, \mathbf{V})$ denotes the inverse Wishart distribution with degrees of freedom v , and positive definite scale matrix $\mathbf{V}$. $\mathcal{IW}(\mathbf{X}; v, \mathbf{V})$ denotes the value of this distribution at a real-valued positive definite matrix $\mathbf{X}$ and it is defined as follows.

$$\mathcal{IW}(\mathbf{X}; v, \mathbf{V}) \triangleq \frac{\text{etr}\left(-\frac{1}{2}\mathbf{X}^{-1}\mathbf{V}\right)}{|\mathbf{X}|^{v/2}},$$

where $\text{etr}(\cdot) \triangleq e^{\text{Tr}(\cdot)}$.

- For the number of measurements $M_k \in \mathbb{Z}^+$, $\mathbf{y}_k^{1:M_k}$ represents the measurement set $\{\mathbf{y}_k^1, \dots, \mathbf{y}_k^{M_k}\}$ at time k .
 - $\mathbf{r}_k$ represents the vector $[r_k^1, \dots, r_k^a]^T$ with size $a \in \mathbb{Z}^+$.
 - $|\mathbf{A}|$ denotes the determinant of the matrix $\mathbf{A}$.
 - $c_{\setminus\phi}$ is a generic constant that denotes the constant terms with respect to variable ϕ in an equation.
 - The subscripts $_{k|k-1}$ and $_{k|k}$ denote the predicted statistics and filtered statistics, respectively.
 - $\text{Tr}(\mathbf{A}) = \sum_{i=1}^n a_{ii}$ where a_{ii} is the i th diagonal element of $\mathbf{A} \in \mathbb{R}^{n \times n}$.
 - The iterate numbers are shown with parenthesized superscripts, e.g., $q^{(i)}(\cdot)$.
-

Table 4.1: Notations (continued)

-
- The expectation with respect to a specific variable is shown with a subscript in the expectation sign, e.g., $\mathbb{E}_{\mathbf{z}}$ will show an expectation with respect $\mathbf{z}_k^{1:M_k}$. When it is clear from the context which variable we are taking the expectation with respect to, we will show the expectation with an overline, e.g., $\overline{(\mathbf{X}_k)^{-1}}$ will denote $\mathbb{E}_{\mathbf{X}}[(\mathbf{X}_k)^{-1}]$.
 - When it is necessary to take expectations with respect to all random variables except for one of them, we will use a backslash “\” in the subscript of the expectation operator, e.g., $\mathbb{E}_{\setminus \mathbf{z}}$ will denote expectation with respect to all random variables except $\mathbf{z}_k^{1:M_k}$.
 - In the derivations, the joint distribution $p(\mathbf{y}_k^{1:M_k}, \mathbf{z}_k^{1:M_k}, \mathbf{r}_k^{1:M_k}, \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L} | \mathcal{Y}_{k-1})$ is abbreviated as $p(\cdot | \mathcal{Y}_{k-1})$ for the sake of brevity.
 - The quadratic forms $\mathbf{x}^T \mathbf{A} \mathbf{x}$ and outer products $\mathbf{x} \mathbf{x}^T$ are written as $\mathbf{x}^T \mathbf{A}(\cdot)$ and $\mathbf{x}(\cdot)^T$, respectively to avoid duplicating long expressions unnecessarily.
-

where $\mathbf{F} \in \mathbb{R}^{n_x \times n_x}$ denotes the state transition matrix and $\mathbf{w}_k \in \mathbb{R}^{n_x}$ denotes zero-mean white Gaussian process noise vector with known SPD covariance matrix $\mathbf{Q}$, i.e., $\mathbf{w}_k \sim \mathcal{N}(\mathbf{w}_k; \mathbf{0}, \mathbf{Q})$.

Suppose at time k , a set of M_k target-originated measurements are captured by the sensor which is denoted as $\mathbf{y}_k^{1:M_k} \triangleq \{\mathbf{y}_k^j\}_{j=1}^{M_k}$. The measurements are assumed to be conditionally i.i.d. and distributed according to the Gaussian mixture given as

$$p(\mathbf{y}_k^j | \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}) = \sum_{\ell=1}^L \pi_k^\ell \mathcal{N}(\mathbf{y}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s \mathbf{X}_k^\ell + \mathbf{R}) \quad (4.3)$$

for $j = 1, \dots, M_k$, where

- $\pi_k^{1:L} \triangleq (\pi_k^1, \pi_k^2, \dots, \pi_k^L)$ is the list of time-varying, unknown and random mixture probabilities π_k^ℓ , $\ell = 1, \dots, L$, satisfying $\pi_k^\ell \geq 0$ and $\sum_{\ell=1}^L \pi_k^\ell = 1$.
- $\mathbf{H}_\ell \in \mathbb{R}^{n_y \times n_x}$, $\ell = 1, \dots, L$ are the measurement matrices defined such that $\mathbf{H}_\ell \mathbf{x}_k \triangleq \mathbf{c}_k^\ell$ where

$$\mathbf{c}_k^\ell \triangleq \begin{cases} \mathbf{x}_k^c, & \ell = 1 \\ \mathbf{x}_k^c + \boldsymbol{\mu}_k^\ell, & \text{otherwise} \end{cases}$$

denotes the center of the ℓ th ellipse.

- $\mathbf{X}_k^{1:L} \triangleq (\mathbf{X}_k^1, \mathbf{X}_k^2, \dots, \mathbf{X}_k^L)$ is the list of sub-object extent matrices $\mathbf{X}_k^\ell$, $\ell = 1, \dots, L$.
- $\mathbf{R} \in \mathbb{R}^{n_y \times n_y}$ is a known SPD measurement noise covariance matrix.
- $s \in \mathbb{R}^+$ is a known positive scaling constant¹.
- Other necessary notations can be found in Table 3.1.

The likelihood of the set of measurements $\mathbf{y}_k^{1:M_k}$ is then given as

$$p(\mathbf{y}_k^{1:M_k} | \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}) = \prod_{j=1}^{M_k} p(\mathbf{y}_k^j | \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}) \quad (4.4a)$$

¹ The value $s = 1/4$ can be used to model uniformly distributed measurements over the ellipse [18, 22].

$$= \prod_{j=1}^{M_k} \sum_{\ell=1}^L \pi_k^\ell \mathcal{N}(\mathbf{y}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s \mathbf{X}_k^\ell + \mathbf{R}). \quad (4.4b)$$

The aim is to find the posterior distribution $p(\mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L} | \mathcal{Y}_k)$ recursively where $\mathcal{Y}_k \triangleq \{\mathbf{y}_t^{1:M_k}\}_{t=0}^k$ denotes the cumulative set of measurements obtained from the sensor up to and including time k . At each time step k , the posterior distribution $p(\mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L} | \mathcal{Y}_k)$ is assumed to be in the form of

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L} | \mathcal{Y}_k) &= \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k}, \mathbf{P}_{k|k}) \\ &\times \left(\prod_{\ell=1}^L \mathcal{IW}(\mathbf{X}_k^\ell; v_{k|k}^\ell, \mathbf{V}_{k|k}^\ell) \right) \\ &\times D(\pi_k^{1:L}; \alpha_{k|k}^{1:L}), \end{aligned} \quad (4.5)$$

where $\mathbf{m}_{k|k}, \mathbf{P}_{k|k}$ are the mean and covariance of the kinematic state vector $\mathbf{x}_k$, respectively. The notations $\mathcal{IW}(\mathbf{X}; v, \mathbf{V})$ and $D(\pi^{1:L}, \alpha^{1:L})$ denote the inverse Wishart and Dirichlet distributions, respectively

$$\mathcal{IW}(\mathbf{X}; v, \mathbf{V}) \triangleq \frac{\text{etr} \left(-\frac{1}{2} \mathbf{X}^{-1} \mathbf{V} \right)}{|\mathbf{X}|^{v/2}}, \quad (4.6a)$$

$$D(\pi^{1:L}; \alpha^{1:L}) \triangleq \prod_{\ell=1}^L (\pi^\ell)^{\alpha^\ell - 1}. \quad (4.6b)$$

The recursive calculation of the posterior $p(\mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L} | \mathcal{Y}_k)$ will be carried out in two steps, namely, the measurement update and time update, which will be investigated separately in the following sections.

4.3 Measurement Update

Suppose now that the predicted distribution $p(\mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L} | \mathcal{Y}_{k-1})$ has the form in (4.5) and is given as

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L} | \mathcal{Y}_{k-1}) &= \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) \\ &\times \left(\prod_{\ell=1}^L \mathcal{IW}(\mathbf{X}_k^\ell; v_{k|k-1}^\ell, \mathbf{V}_{k|k-1}^\ell) \right) \\ &\times D(\pi_k^{1:L}; \alpha_{k|k-1}^{1:L}). \end{aligned} \quad (4.7)$$

In the measurement update, we would like to update the predicted distribution in (4.7) with the likelihood in (4.4b). Unfortunately, such an update is not analytically tractable, and furthermore, it would not result in a posterior in the form (4.5). In order to ensure analytical tractability and to preserve the form of the posterior, first, we are going to define some latent variables and then resort to variational approximation.

We first define the latent variables $\mathbf{z}_k^j$, $j = 1, \dots, M_k$, which represent the noise-free measurements as in [60], as

$$p(\mathbf{z}_k^j | \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}) = \sum_{\ell=1}^L \pi_k^\ell \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s \mathbf{X}_k^\ell), \quad (4.8a)$$

$$p(\mathbf{y}_k^j | \mathbf{z}_k^j) = \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}). \quad (4.8b)$$

Note that the conditional joint distribution for $\mathbf{y}_k^j, \mathbf{z}_k^j$ is given as

$$\begin{aligned} p(\mathbf{y}_k^j, \mathbf{z}_k^j | \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}) &= \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) \\ &\times \sum_{\ell=1}^L \pi_k^\ell \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s \mathbf{X}_k^\ell). \end{aligned} \quad (4.9)$$

We also define the association/responsibility vector $\mathbf{r}_k^j$ for each measurement $\mathbf{y}_k^j$ (or $\mathbf{z}_k^j$), which is a binary vector of size L defined as

$$\mathbf{r}_k^j \triangleq [r_k^{j,1} \quad r_k^{j,2} \quad \dots \quad r_k^{j,L}]^T \quad (4.10)$$

where the elements $r_k^{j,\ell} \in \{0, 1\}$, $\ell = 1, \dots, L$, which are called responsibilities in the literature [8, Ch. 10]. These binary variables satisfy

$$\sum_{\ell=1}^L r_k^{j,\ell} = 1. \quad (4.11)$$

Note that, with these properties, the elements $r_k^{j,\ell}$, $\ell = 1, \dots, L$, are all equal to zero except for one of them which is unity. The index ℓ^* for which $r_k^{j,\ell}$ is equal to unity, i.e., $r_k^{j,\ell^*} = 1$, is the index of the sub-object which the measurement $\mathbf{y}_k^j$ (or $\mathbf{z}_k^j$) is associated to. The distribution of $r_k^{j,\ell}$ is defined as

$$P\{r_k^{j,\ell} = 1 | \pi_k^{1:L}\} \triangleq \pi_k^\ell, \quad \text{for } \ell = 1, \dots, L, \quad (4.12)$$

for $j = 1, \dots, M_k$. The expression (4.12) can be written as

$$P\{\mathbf{r}_k^j | \pi_k^{1:L}\} \triangleq \prod_{\ell=1}^L (\pi_k^\ell)^{r_k^{j,\ell}}. \quad (4.13)$$

Note that given the association vector $\mathbf{r}_k^j$, the noisy and noiseless measurements $\mathbf{y}_k^j$ and $\mathbf{z}_k^j$ are distributed as

$$p(\mathbf{y}_k^j | \mathbf{r}_k^j, \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}) = \mathcal{N}(\mathbf{y}_k^j; \mathbf{H}_{\ell^*} \mathbf{x}_k, s \mathbf{X}_k^{\ell^*} + \mathbf{R}), \quad (4.14a)$$

$$p(\mathbf{z}_k^j | \mathbf{r}_k^j, \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}) = \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_{\ell^*} \mathbf{x}_k, s \mathbf{X}_k^{\ell^*}), \quad (4.14b)$$

for $j = 1, \dots, M_k$ where ℓ^* is the index for which the element $r_k^{j,\ell}$ is equal to unity, i.e., $r_k^{j,\ell^*} = 1$. The expressions (4.14) can conveniently be written as

$$p(\mathbf{y}_k^j | \mathbf{r}_k^j, \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}) = \prod_{\ell=1}^L \mathcal{N}^{r_k^{j,\ell}}(\mathbf{y}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s \mathbf{X}_k^\ell + \mathbf{R}), \quad (4.15a)$$

$$p(\mathbf{z}_k^j | \mathbf{r}_k^j, \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}) = \prod_{\ell=1}^L \mathcal{N}^{r_k^{j,\ell}}(\mathbf{z}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s \mathbf{X}_k^\ell), \quad (4.15b)$$

for $j = 1, \dots, M_k$. Using these expressions, we can write the conditional joint distribution for $\mathbf{y}_k^j, \mathbf{z}_k^j, \mathbf{r}_k^j$ as

$$\begin{aligned} p(\mathbf{y}_k^j, \mathbf{z}_k^j, \mathbf{r}_k^j | \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}) &= p(\mathbf{y}_k^j | \mathbf{z}_k^j) \\ &\quad \times p(\mathbf{z}_k^j | \mathbf{r}_k^j, \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}) \\ &\quad \times P\{\mathbf{r}_k^j | \pi_k^{1:L}\}, \end{aligned} \quad (4.16a)$$

$$= \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) \prod_{\ell=1}^L \mathcal{N}^{r_k^{j,\ell}}(\mathbf{z}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s \mathbf{X}_k^\ell) \prod_{\ell=1}^L (\pi_k^\ell)^{r_k^{j,\ell}}, \quad (4.16b)$$

$$= \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) \prod_{\ell=1}^L (\pi_k^\ell)^{r_k^{j,\ell}} \mathcal{N}^{r_k^{j,\ell}}(\mathbf{z}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s \mathbf{X}_k^\ell). \quad (4.16c)$$

The overall conditional joint distribution for $\mathbf{y}_k^{1:M_k}, \mathbf{z}_k^{1:M_k}, \mathbf{r}_k^{1:M_k}$ can then be written as

$$\begin{aligned} p(\mathbf{y}_k^{1:M_k}, \mathbf{z}_k^{1:M_k}, \mathbf{r}_k^{1:M_k} | \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}) \\ = \prod_{j=1}^{M_k} \left(\mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) \prod_{\ell=1}^L (\pi_k^\ell)^{r_k^{j,\ell}} \mathcal{N}^{r_k^{j,\ell}}(\mathbf{z}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s \mathbf{X}_k^\ell) \right). \end{aligned} \quad (4.17)$$

Since the posterior for $\mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}$ given $\mathcal{Y}_k$ is intractable, we aim for approximating the joint posterior for $\mathbf{z}_k^{1:M_k}, \mathbf{r}_k^{1:M_k}, \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}$ given $\mathcal{Y}_k$ in the following form,

$$p(\mathbf{z}_k^{1:M_k}, \mathbf{r}_k^{1:M_k}, \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L} | \mathcal{Y}_k) \approx q_{\mathbf{z}}(\mathbf{z}_k^{1:M_k}) q_{\mathbf{r}}(\mathbf{r}_k^{1:M_k}) q_{\mathbf{x}}(\mathbf{x}_k) q_{\mathbf{X}}(\mathbf{X}_k^{1:L}) q_{\pi}(\pi_k^{1:L}). \quad (4.18)$$

We calculate the terms of the approximation above iteratively using variational approximation with the following true joint density

$$\begin{aligned} & p(\mathbf{y}_k^{1:M_k}, \mathbf{z}_k^{1:M_k}, \mathbf{r}_k^{1:M_k}, \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L} | \mathcal{Y}_{k-1}) \\ &= p(\mathbf{y}_k^{1:M_k}, \mathbf{z}_k^{1:M_k}, \mathbf{r}_k^{1:M_k} | \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L}) \\ & \quad \times p(\mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L} | \mathcal{Y}_{k-1}), \\ &= \prod_{j=1}^{M_k} \left(\mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) \prod_{\ell=1}^L (\pi_k^\ell)^{r_k^{j,\ell}} \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s \mathbf{X}_k^\ell) \right) \\ & \quad \times \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) \\ & \quad \times \left(\prod_{\ell=1}^L \mathcal{IW}(\mathbf{X}_k^\ell; v_{k|k-1}^\ell, \mathbf{V}_{k|k-1}^\ell) \right) \\ & \quad \times D(\pi_k^{1:L}; \{\alpha_{k|k-1}^\ell\}_{\ell=1}^L). \end{aligned} \quad (4.19)$$

The logarithm of the joint density above is given as

$$\begin{aligned} & \log p(\mathbf{y}_k^{1:M_k}, \mathbf{z}_k^{1:M_k}, \mathbf{r}_k^{1:M_k}, \mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L} | \mathcal{Y}_{k-1}) \\ &= \sum_{j=1}^{M_k} \left(\log \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) \right. \\ & \quad \left. + \sum_{\ell=1}^L r_k^{j,\ell} \left(\log \pi_k^\ell + \log \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s \mathbf{X}_k^\ell) \right) \right) \\ & \quad + \log \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) \\ & \quad + \sum_{\ell=1}^L \log \mathcal{IW}(\mathbf{X}_k^\ell; v_{k|k-1}^\ell, \mathbf{V}_{k|k-1}^\ell) \\ & \quad + \log D(\pi_k^{1:L}; \{\alpha_{k|k-1}^\ell\}_{\ell=1}^L). \end{aligned} \quad (4.20)$$

With this log-distribution, the factors $q_{\mathbf{z}}(\cdot)$, $q_{\mathbf{r}}(\cdot)$, $q_{\mathbf{x}}(\cdot)$, $q_{\mathbf{X}}(\cdot)$, and $q_{\pi}(\cdot)$ can be calcu-

lated using variational Bayes approach [8, Ch. 10] as

$$q_{\mathbf{z}}(\mathbf{z}_k^{1:M_k}) = \prod_{j=1}^{M_k} \mathcal{N}(\mathbf{z}_k^j; \hat{\mathbf{z}}_{k|k}^j, \mathbf{P}_{k|k}^{z,j}), \quad (4.21a)$$

$$q_{\mathbf{r}}(\mathbf{r}_k^{1:M}) = \prod_{j=1}^{M_k} \prod_{\ell=1}^L (\gamma_{k|k}^{j,\ell})^{r_k^{j,\ell}}, \quad (4.21b)$$

$$q_{\mathbf{x}}(\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k}, \mathbf{P}_{k|k}), \quad (4.21c)$$

$$q_{\mathbf{X}}(\mathbf{X}_k^{1:L}) = \prod_{\ell=1}^L \mathcal{IW}(\mathbf{X}_k^\ell; v_{k|k}^\ell, \mathbf{V}_{k|k}^\ell), \quad (4.21d)$$

$$q_{\pi}(\pi_k^{1:L}) = D(\pi_k^{1:L}; \alpha_{k|k}^{1:L}), \quad (4.21e)$$

where the parameters $\hat{\mathbf{z}}_{k|k}^j$, $\mathbf{P}_{k|k}^{z,j}$, $\gamma_{k|k}^{j,\ell}$, $\mathbf{m}_{k|k}$, $\mathbf{P}_{k|k}$, $v_{k|k}^\ell$, $\mathbf{V}_{k|k}^\ell$, $\alpha_{k|k}^\ell$, $j = 1, \dots, M_k$, $\ell = 1, \dots, L$, are found iteratively. Once the updated distributions in (4.21) are obtained we can approximate the updated posterior $p(\mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L} | \mathcal{Y}_k)$ as

$$p(\mathbf{x}_k, \mathbf{X}_k^{1:L}, \pi_k^{1:L} | \mathcal{Y}_k) \approx q_{\mathbf{x}}(\mathbf{x}_k) q_{\mathbf{X}}(\mathbf{X}_k^{1:L}) q_{\pi}(\pi_k^{1:L}), \quad (4.22)$$

which is in the form (4.5) as required. The distributions in (4.21) are calculated iteratively. The expressions for the corresponding iterations are given in the following subsections. The detailed derivations of these expressions and the required expectations are given in Appendix .1.1. A summary of the resulting iterative update procedure can be found in Algorithm 4.1.

4.3.1 Calculation of $q_{\mathbf{z}}^{(i+1)}(\cdot)$

The $(i+1)$ th iterate $q_{\mathbf{z}}^{(i+1)}(\cdot)$ of $q_{\mathbf{z}}(\cdot)$ is given as

$$q_{\mathbf{z}}^{(i+1)}(\mathbf{z}_k^{1:M_k}) = \prod_{j=1}^{M_k} \mathcal{N}(\mathbf{z}_k^j; \hat{\mathbf{z}}_{k|k}^{j,(i+1)}, \mathbf{P}_{k|k}^{z,j,(i+1)}), \quad (4.23)$$

where

$$\hat{\mathbf{z}}_{k|k}^{j,(i+1)} \triangleq \mathbf{P}_{k|k}^{z,j,(i+1)} \left(\mathbf{R}^{-1} \mathbf{y}_k^j + \sum_{\ell=1}^L \overline{r_k^{j,\ell} (s \mathbf{X}_k^\ell)^{-1}} \mathbf{H}_\ell \bar{\mathbf{x}}_k \right), \quad (4.24a)$$

$$\mathbf{P}_{k|k}^{z,j,(i+1)} \triangleq \left(\mathbf{R}^{-1} + \sum_{\ell=1}^L \overline{r_k^{j,\ell} (s \mathbf{X}_k^\ell)^{-1}} \right)^{-1}, \quad (4.24b)$$

for $j = 1, \dots, M_k$.

4.3.2 Calculation of $q_{\mathbf{r}}^{(i+1)}(\cdot)$

The $(i + 1)$ th iterate $q_{\mathbf{r}}^{(i+1)}(\cdot)$ of $q_{\mathbf{r}}(\cdot)$ is given as

$$q_{\mathbf{r}}^{(i+1)}(\mathbf{r}_k^{1:M_k}) = \prod_{j=1}^{M_k} \prod_{\ell=1}^L (\gamma_{k|k}^{j,\ell,(i+1)})^{r_k^{j,\ell}}, \quad (4.25)$$

where

$$\gamma_{k|k}^{j,\ell,(i+1)} \triangleq \frac{\tilde{\gamma}_{k|k}^{j,\ell,(i+1)}}{\sum_{\ell=1}^L \tilde{\gamma}_{k|k}^{j,\ell,(i+1)}}, \quad (4.26a)$$

$$\begin{aligned} \tilde{\gamma}_{k|k}^{j,\ell,(i+1)} \triangleq & \exp \left(\overline{\log \pi_k^\ell} - \frac{1}{2} \overline{\log |\mathbf{X}_k^\ell|} \right. \\ & \left. - \frac{1}{2} \text{Tr} \left(\left(s \mathbf{X}_k^\ell \right)^{-1} \left(\mathbf{z}_k^j - \mathbf{H}_{\ell} \mathbf{x}_k \right) \left(\cdot \right)^T \right) \right), \end{aligned} \quad (4.26b)$$

for $j = 1, \dots, M_k, \ell = 1, \dots, L$.

4.3.3 Calculation of $q_{\mathbf{x}}^{(i+1)}(\cdot)$

The $(i + 1)$ th iterate $q_{\mathbf{x}}^{(i+1)}(\cdot)$ of $q_{\mathbf{x}}(\cdot)$ is given as

$$q_{\mathbf{x}}^{(i+1)}(\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k}^{(i+1)}, \mathbf{P}_{k|k}^{(i+1)}), \quad (4.27)$$

where

$$\mathbf{m}_{k|k}^{(i+1)} = \mathbf{P}_{k|k}^{(i+1)} (\mathbf{P}_{k|k-1}^{-1} \mathbf{m}_{k|k-1} + \mathbf{H}_{1:L}^T \mathbf{\Lambda}_{1:L} \mathbf{u}_k^{1:L}), \quad (4.28a)$$

$$\mathbf{P}_{k|k}^{(i+1)} = (\mathbf{P}_{k|k-1}^{-1} + \mathbf{H}_{1:L}^T \mathbf{\Lambda}_{1:L} \mathbf{H}_{1:L})^{-1}, \quad (4.28b)$$

$$\mathbf{u}_k^{1:L} \triangleq [(\mathbf{u}_k^1)^T \quad (\mathbf{u}_k^2)^T \quad \dots \quad (\mathbf{u}_k^L)^T]^T, \quad (4.28c)$$

$$\mathbf{H}_{1:L} \triangleq [\mathbf{H}_1^T \quad \mathbf{H}_2^T \quad \dots \quad \mathbf{H}_L^T]^T, \quad (4.28d)$$

$$\mathbf{\Lambda}_{1:L} \triangleq \text{blkdiag}(\mathbf{\Lambda}_k^1, \mathbf{\Lambda}_k^2, \dots, \mathbf{\Lambda}_k^L). \quad (4.28e)$$

Here, we have

$$\mathbf{u}_k^\ell \triangleq \frac{\sum_{j=1}^{M_k} \overline{r_k^{j,\ell} \mathbf{z}_k^j}}{\sum_{j=1}^{M_k} \overline{r_k^{j,\ell}}}, \quad (4.28f)$$

$$\mathbf{\Lambda}_k^\ell \triangleq \overline{(s \mathbf{X}_k^\ell)^{-1}} \sum_{j=1}^{M_k} \overline{r_k^{j,\ell}}, \quad (4.28g)$$

for $\ell = 1, \dots, L$.

4.3.4 Calculation of $q_{\mathbf{X}}^{(i+1)}(\cdot)$

The $(i + 1)$ th iterate $q_{\mathbf{X}}^{(i+1)}(\cdot)$ of $q_{\mathbf{X}}(\cdot)$ is given as

$$q_X^{(i+1)}(\mathbf{X}_k^{1:L}) = \prod_{\ell=1}^L \mathcal{IW}(\mathbf{X}_k^\ell; v_{k|k}^{\ell,(i+1)}, \mathbf{V}_{k|k}^{\ell,(i+1)}), \quad (4.29)$$

where

$$v_{k|k}^{\ell,(i+1)} \triangleq v_{k|k-1}^\ell + \sum_{j=1}^{M_k} \overline{r_k^{j,\ell}}, \quad (4.30a)$$

$$\mathbf{V}_{k|k}^{\ell,(i+1)} \triangleq \mathbf{V}_{k|k-1}^\ell + \frac{1}{S} \sum_{j=1}^M \overline{r_k^{j,\ell} (\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k) (\cdot)^T}, \quad (4.30b)$$

for $\ell = 1, \dots, L$.

4.3.5 Calculation of $q_{\pi}^{(i+1)}(\cdot)$

The $(i + 1)$ th iterate $q_{\pi}^{(i+1)}(\cdot)$ of $q_{\pi}(\cdot)$ is given as

$$q_{\pi}^{(i+1)}(\pi_k^{1:L}) = D(\pi_k^{1:L}; \{\alpha_{k|k}^{\ell,(i+1)}\}_{\ell=1}^L), \quad (4.31)$$

where

$$\alpha_{k|k}^{\ell,(i+1)} \triangleq \alpha_{k|k-1}^\ell + \sum_{j=1}^{M_k} \overline{r_k^{j,\ell}}, \quad (4.32)$$

for $\ell = 1, \dots, L$.

Algorithm 4.1 Measurement Update for ME-ETT Model

- 1: **Given:** $\mathbf{m}_{k|k-1}$, $\mathbf{P}_{k|k-1}$, $\{v_{k|k-1}^\ell, \mathbf{V}_{k|k-1}^\ell, \alpha_{k|k-1}^\ell\}_{\ell=1}^L$ and $\mathbf{y}_k^{1:M_k}$, calculate $\mathbf{m}_{k|k}$, $\mathbf{P}_{k|k}$, $\{v_{k|k}^\ell, \mathbf{V}_{k|k}^\ell, \alpha_{k|k}^\ell\}_{\ell=1}^L$ as follows.
- 2: **Initialization at time $k = 0$:**
- 3: $\mathbf{m}_{k|k}^{(0)} \leftarrow \mathbf{m}_{k|k-1}$
- 4: $\mathbf{P}_{k|k}^{(0)} \leftarrow \mathbf{P}_{k|k-1}$
- 5: **for** $\ell = 1, \dots, L$ **do**
- 6: $v_{k|k}^{\ell,(0)} \leftarrow v_{k|k-1}^\ell$

```

7:    $\mathbf{V}_{k|k}^{\ell,(0)} \leftarrow \mathbf{V}_{k|k-1}^\ell$ 
8:    $\alpha_{k|k}^{\ell,(0)} \leftarrow \alpha_{k|k-1}^\ell$ 
9:   for  $j = 1, \dots, M_k$  do
10:     $\gamma_{k|k}^{j,\ell,(0)} \leftarrow \frac{\alpha_{k|k-1}^\ell}{\sum_{\ell'=1}^L \alpha_{k|k-1}^{\ell'}}$ 
11:   end for
12: end for
13: for  $j = 1, \dots, M_k$  do
14:    $\mathbf{z}_{k|k}^{j,(0)} \leftarrow \mathbf{y}_k^j$ 
15:    $\mathbf{P}_{k|k}^{z,j,(0)} \leftarrow \mathbf{R}$ 
16: end for
17: Iterations:
18:  $\mathbf{H}_{1:L} = [\mathbf{H}_1^T \ \mathbf{H}_2^T \ \dots \ \mathbf{H}_L^T]^T$ 
19: for  $i = 0, \dots, i_{max} - 1$  do
20:   for  $j = 1, \dots, M_k$  do
21:    for  $\ell = 1, \dots, L$  do
22:      $\mathbf{W}_{j\ell}^{(i)} \triangleq (\mathbf{z}_{k|k}^{j,(i)} - \mathbf{H}_\ell \mathbf{m}_{k|k}^{(i)}) (\mathbf{z}_{k|k}^{j,(i)} - \mathbf{H}_\ell \mathbf{m}_{k|k}^{(i)})^T + \mathbf{P}_{k|k}^{z,j,(i)} + \mathbf{H}_\ell \mathbf{P}_{k|k}^{(i)} \mathbf{H}_\ell^T$ 
23:      $\tilde{\gamma}_{k|k}^{j,\ell,(i+1)} = \exp \left( \psi(\alpha_{k|k}^{\ell,(i)}) - \frac{1}{2} \log |\mathbf{V}_{k|k}^{\ell,(i)}| + \frac{1}{2} \sum_{d=1}^{n_y} \psi \left( \frac{v_{k|k}^{\ell,(i)} - n_y - d}{2} \right) \right.$ 
24:      $\left. - \frac{v_{k|k}^{\ell,(i)} - n_y - 1}{2} \text{Tr} \left( (s \mathbf{V}_{k|k}^{\ell,(i)})^{-1} \mathbf{W}_{j\ell}^{(i)} \right) \right)$ 
25:    end for
26:    for  $\ell = 1, \dots, L$  do
27:      $\gamma_{k|k}^{j,\ell,(i+1)} = \frac{\tilde{\gamma}_{k|k}^{j,\ell,(i+1)}}{\sum_{\ell'=1}^L \tilde{\gamma}_{k|k}^{j,\ell',(i+1)}}$ 
28:    end for
29:   for  $j = 1, \dots, M_k$  do
30:     $\mathbf{P}_{k|k}^{z,j,(i+1)} = \left( \mathbf{R}^{-1} + \sum_{\ell=1}^L \left[ \gamma_{k|k}^{j,\ell,(i)} (v_{k|k}^{\ell,(i)} - n_y - 1) (s \mathbf{V}_{k|k}^{\ell,(i)})^{-1} \right] \right)^{-1}$ 
31:     $\mathbf{z}_{k|k}^{j,(i+1)} = \mathbf{P}_{k|k}^{z,j,(i+1)} \left( \mathbf{R}^{-1} \mathbf{y}_k^j \right.$ 
32:     $\left. + \sum_{\ell=1}^L \left[ \gamma_{k|k}^{j,\ell,(i)} (v_{k|k}^{\ell,(i)} - n_y - 1) (s \mathbf{V}_{k|k}^{\ell,(i)})^{-1} \mathbf{H}_\ell \mathbf{m}_{k|k}^{(i)} \right] \right)$ 
33:   end for

```

```

33: for  $\ell = 1, \dots, L$  do
34:    $v_{k|k}^{\ell, (i+1)} = v_{k|k-1}^\ell + \sum_{j=1}^{M_k} \gamma_{k|k}^{j, \ell, (i)}$ 
35:    $\mathbf{V}_{k|k}^{\ell, (i+1)} = \mathbf{V}_{k|k-1}^\ell + \frac{1}{s} \sum_{j=1}^{M_k} [\gamma_{k|k}^{j, \ell, (i)} \mathbf{W}_{j\ell}^{(i)}]$ 
36:    $\alpha_{k|k}^{\ell, (i+1)} = \alpha_{k|k-1}^\ell + \sum_{j=1}^{M_k} \gamma_{k|k}^{j, \ell, (i)}$ 
37:    $\mathbf{u}_k^\ell = \frac{\sum_{j=1}^{M_k} [\gamma_{k|k}^{j, \ell, (i)} \mathbf{z}_{k|k}^{j, (i)}]}{\sum_{j=1}^{M_k} \gamma_{k|k}^{j, \ell, (i)}}$ 
38:    $\Lambda_k^\ell = (v_{k|k}^{\ell, (i)} - n_y - 1) (s \mathbf{V}_{k|k}^{\ell, (i)})^{-1} \sum_{j=1}^{M_k} \gamma_{k|k}^{j, \ell, (i)}$ 
39: end for
40:  $\mathbf{u}_k^{1:L} = [ (\mathbf{u}_k^1)^T \ (\mathbf{u}_k^2)^T \ \dots \ (\mathbf{u}_k^L)^T ]^T$ 
41:  $\Lambda_{1:L} = \text{blkdiag}(\Lambda_k^1, \Lambda_k^2, \dots, \Lambda_k^L)$ 
42:  $\mathbf{P}_{k|k}^{(i+1)} = \left( \mathbf{P}_{k|k-1}^{-1} + \mathbf{H}_{1:L}^T \Lambda_{1:L} \mathbf{H}_{1:L} \right)^{-1}$ 
43:  $\mathbf{m}_{k|k}^{(i+1)} = \mathbf{P}_{k|k}^{(i+1)} \left( \mathbf{P}_{k|k-1}^{-1} \mathbf{m}_{k|k-1} + \mathbf{H}_{1:L}^T \Lambda_{1:L} \mathbf{u}_k^{1:L} \right)$ 
44: end for
45: Set final estimates:
46:  $\mathbf{m}_{k|k} = \mathbf{m}_{k|k}^{(i_{\max})}$ 
47:  $\mathbf{P}_{k|k} = \mathbf{P}_{k|k}^{(i_{\max})}$ 
48: for  $\ell = 1, \dots, L$  do
49:    $v_{k|k}^\ell = v_{k|k}^{\ell, (i_{\max})}$ 
50:    $\mathbf{V}_{k|k}^\ell = \mathbf{V}_{k|k}^{\ell, (i_{\max})}$ 
51:    $\alpha_{k|k}^\ell = \alpha_{k|k}^{\ell, (i_{\max})}$ 
52: end for

```

4.4 Time Update

With the target dynamics given in (4.2), the time update of the kinematic state is performed following the regular Kalman filter time update equations

$$\mathbf{m}_{k|k-1} = \mathbf{F} \mathbf{m}_{k-1|k-1}, \quad (4.33a)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F} \mathbf{P}_{k-1|k-1} \mathbf{F}^T + \mathbf{Q}. \quad (4.33b)$$

In this proposition, we assume that the extent state of the target is slowly varying with unknown dynamics, which is usually the case in various target tracking applications.

The parameters of the inverse Wishart distribution and the Dirichlet distribution of the extent model are updated with forgetting factors as proposed in [62] for exponential family of distributions,

$$v_{k|k-1}^\ell = \lambda_{IW} v_{k-1|k-1}^\ell, \quad (4.34a)$$

$$\mathbf{V}_{k|k-1}^\ell = \lambda_{IW} \mathbf{V}_{k-1|k-1}^\ell, \quad \text{for } \ell = 1, \dots, L. \quad (4.34b)$$

The forgetting factor is used to obtain the maximum entropy prediction density of the extent states when the dynamics of the extent state is slowly varying and unknown [62, Theorem 1]. Similarly, the sufficient statistics of the Dirichlet distributed mixture weights $\pi_k^{1:L}$ are propagated with a forgetting factor λ_D ,

$$\alpha_{k|k-1}^\ell = \lambda_D \alpha_{k-1|k-1}^\ell \quad \text{for } \ell = 1, \dots, L. \quad (4.35)$$

If the true parameter evolution is slowly varying, the time update equations (4.34) and (4.35) will not underestimate the uncertainty by maximizing the entropy. Note that, for the special case of stationary parameters forgetting factors are set to 1, i.e., $\lambda_{IW} = \lambda_D = 1$.

The proposed algorithm is also versatile to perform under alternative random matrix time update schemes.

4.5 Experimental Results

In this section, we will demonstrate the performance of the proposed algorithm in various experiments and compare its performance with the algorithms presented in [18, 39, 45, 90]. In the sequel, we will denote the proposed algorithm as VB (Variational Bayes), the method in [39] as VPF (Variational Particle Filter), and the method in [45] as LL (denoting the initials of the authors' surnames).

To assess the performance of the algorithms, we consider the Intersection-Over-Union (IOU) similarity measure [17, 24, 82] together with Gaussian Wasserstein (GW) distance [19, 91] for the extent estimates and the root-mean-square error (RMSE) for the center of the ellipses. The RMSE between the true and the estimated center of the ℓ th

ellipse is defined as

$$\text{RMSE}(\mathbf{c}_{1:N}^{\ell, \text{true}}, \hat{\mathbf{c}}_{1:N}^{\ell}) = \sqrt{\frac{1}{N} \sum_{k=1}^N \|\mathbf{c}_k^{\ell, \text{true}} - \hat{\mathbf{c}}_k^{\ell}\|_2^2}, \quad (4.36)$$

where N is the number of time steps in a single Monte Carlo (MC) run. The true and estimated center of the ℓ th ellipse at time k are denoted as $\mathbf{c}_k^{\ell, \text{true}}$ and $\hat{\mathbf{c}}_k^{\ell}$, respectively.

The IOU measure between the estimated extent and the true extent of the ℓ th sub-object is calculated as

$$\text{GW}(\mathbf{X}_k^{\ell, \text{true}}, \hat{\mathbf{X}}_k^{\ell}) = \frac{\text{area}(\mathbf{X}_k^{\ell, \text{true}} \cap \hat{\mathbf{X}}_k^{\ell})}{\text{area}(\mathbf{X}_k^{\ell, \text{true}} \cup \hat{\mathbf{X}}_k^{\ell})} \in [0, 1], \quad (4.37a)$$

$$\text{GW}(\mathbf{X}_{1:N}^{\ell, \text{true}}, \hat{\mathbf{X}}_{1:N}^{\ell}) = \frac{1}{N} \sum_{k=1}^N \text{GW}(\mathbf{X}_k^{\ell, \text{true}}, \hat{\mathbf{X}}_k^{\ell}), \quad (4.37b)$$

where $\mathbf{X}_k^{\ell, \text{true}}$ and $\hat{\mathbf{X}}_k^{\ell} \triangleq \frac{\mathbf{V}_{k|k}^{\ell}}{v_{k|k}^{\ell} - 2n_y - 2}$ are the true and estimated extent matrices of the ℓ th sub-object at time k , respectively. Note that IOU takes values between 0 and 1, where 1 corresponds to the perfect match, while 0 indicates no intersection between the true and the estimated extents. The GW distance [91] between the estimated extent and the true extent of the ℓ th sub-object can be expressed as

$$\begin{aligned} \text{GW}(\mathbf{c}_k^{\ell, \text{true}}, \mathbf{X}_k^{\ell, \text{true}}, \hat{\mathbf{c}}_k^{\ell}, \hat{\mathbf{X}}_k^{\ell})^2 &\triangleq \|\mathbf{c}_k^{\ell, \text{true}} - \hat{\mathbf{c}}_k^{\ell}\|_2^2 \\ &+ \text{Tr} \left[\mathbf{X}_k^{\ell, \text{true}} + \hat{\mathbf{X}}_k^{\ell} - 2(\mathbf{X}_k^{\ell, \text{true}})^{\frac{1}{2}} \hat{\mathbf{X}}_k^{\ell} (\mathbf{X}_k^{\ell, \text{true}})^{\frac{1}{2}} \right]^{\frac{1}{2}}, \end{aligned} \quad (4.38a)$$

$$\begin{aligned} \text{GW}(\mathbf{c}_{1:N}^{\ell, \text{true}}, \mathbf{X}_{1:N}^{\ell, \text{true}}, \hat{\mathbf{c}}_{1:N}^{\ell}, \hat{\mathbf{X}}_{1:N}^{\ell}) \\ = \frac{1}{N} \sum_{k=1}^N \text{GW}(\mathbf{c}_k^{\ell, \text{true}}, \mathbf{X}_k^{\ell, \text{true}}, \hat{\mathbf{c}}_k^{\ell}, \hat{\mathbf{X}}_k^{\ell}). \end{aligned} \quad (4.38b)$$

The comparison metrics above are formulated for a single MC run. In the sequel, all simulations are performed 100 times with different realizations of the measurement noise at each run. The presented numbers in the simulations are the averages of the 100 MC runs.

4.5.1 Simulations

The first simulation consists of two co-centered elliptical objects moving according to the near constant velocity model with parameters given in Table 4.2.

Throughout the simulation, 7 measurements are generated from each object per time step.

We compare the performances of VB, VPF (with $N=100$ particles), and LL algorithms over 100 MC runs. An example MC run is illustrated in Fig. 4.3. The comparison metrics and the computation time of the algorithms are given in Table 4.3. VB algorithm estimates both the kinematic and extent states better than other approaches in terms of RMSE, IOU, and GW distance.

Note that the number of association events in the LL algorithm grows exponentially as the number of the sub-objects and measurements increases [45]. Hence the LL algorithm's computation time is significantly higher than the other methods, as shown in Table 4.3. Being a sequential Monte Carlo method, VPF algorithm is also computationally costly.

On the other hand, the computation time of VB is linear in both number of measurements M_k and number of sub-objects L . The computational complexity of the proposed solution is $\mathcal{O}(LM_k)$ at time k . Consequently, the computation time of the VB algorithm is significantly lower than the alternatives as shown in Table 4.3.

In the second simulation scenario, we have a target with a shape similar to an airplane (see: Fig. 4.4.). The object performs a constant velocity motion on a straight line.

The relevant parameters of the scenario are given in Table 4.4. Throughout the simulation, the number of measurements is set to 2 for each sub-object, yielding 8 measurements in total per time step. The performance evaluation metrics are given in Table 4.5. The VB algorithm outperforms LL and VPF algorithms in terms of accuracy in the extent estimates and provides results which exhibit smaller variation over time.

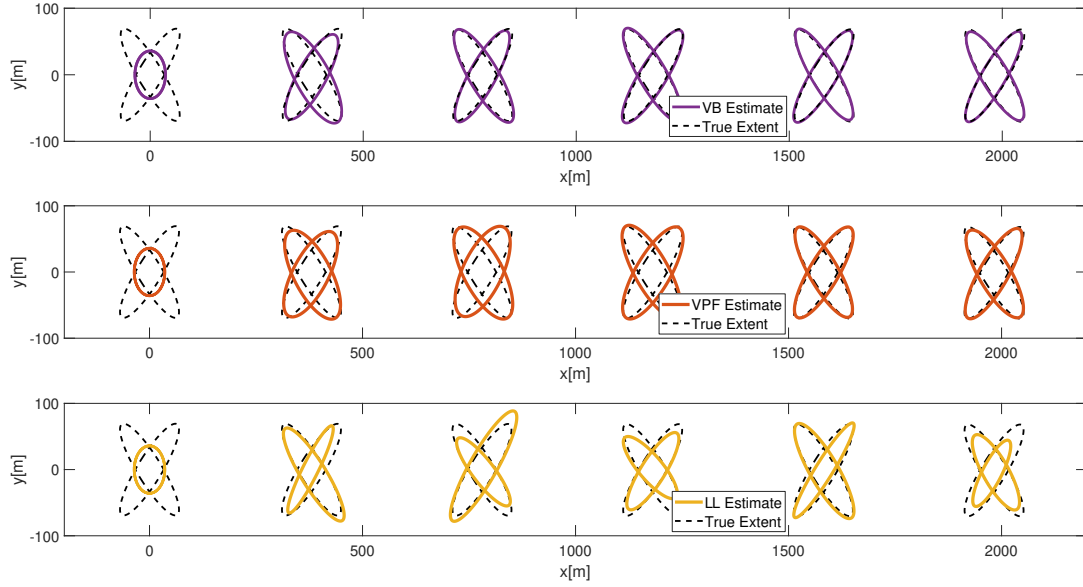


Figure 4.3: The estimation results for the first scenario where two co-centered objects move along a straight line. The extent estimates of VB, VPF, and LL are shown in purple, orange, and yellow lines, respectively. The ground truth is depicted by black dashed lines.

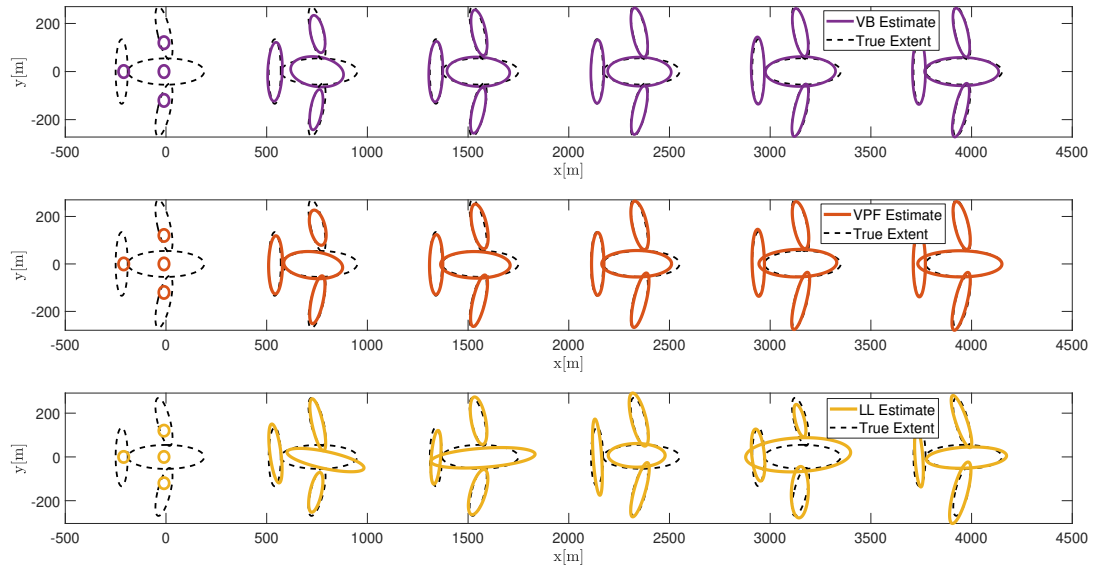


Figure 4.4: The estimation results for the second scenario where four sub-objects with distinct extents move along a straight line. The extent estimates of VB, VPF, and LL are shown in purple, orange, and yellow lines, respectively. The ground truth is depicted by black dashed lines.

Table 4.2: The simulation parameters for the first scenario.

$\mathbf{X}_{\text{true}}$	$\begin{bmatrix} 800 & 700 \\ 700 & 800 \end{bmatrix}$	$\begin{bmatrix} 800 & -700 \\ -700 & 800 \end{bmatrix}$
$\mathbf{m}_0$	$[0, 0, 200, 0, 0, 0]^T$	
$\mathbf{P}_0$	$\text{blkdiag}(50\mathbf{I}_2, 10\mathbf{I}_4)$	
$\mathbf{V}_0$	$1500\mathbf{I}_2$	
v_0	10	
$\mathbf{F}_c$	$\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes \mathbf{I}_2$	
$\mathbf{F}_{\mu,\ell}$	$\mathbf{I}_2$	
$\mathbf{F}$	$\text{blkdiag}(\mathbf{F}_c, \mathbf{F}_{\mu,2})$	
$\mathbf{Q}_c$	$\sigma^2 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \otimes \mathbf{I}_2$	
$\mathbf{Q}_{\mu,\ell}$	$\mathbf{I}_2$	
$\mathbf{Q}$	$\text{blkdiag}(\mathbf{Q}_c, \mathbf{Q}_{\mu,2})$	
$\mathbf{R}$	$10\mathbf{I}_2$	
σ^2	0.1	
T	0.1	

Table 4.3: The RMSE, IOU, GW values and the computation times for the first scenario.

	Sub-obj	IOU	RMSE	GW	Comp. Time Per Time Step
LL	1 st	0.70	3.89	9.68	13.96 s
	2 nd	0.70	4.26	9.69	
VPF	1 st	0.70	1.95	8.64	298.56 ms
	2 nd	0.70	2.12	8.57	
VB	1 st	0.77	1.64	5.06	25.91 ms
	2 nd	0.77	1.92	5.08	

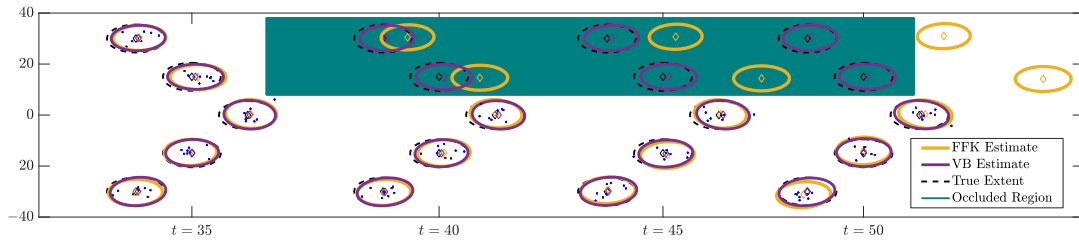


Figure 4.5: The extent estimates of VB (purple solid line) and FFK (yellow solid line) together with the ground truth (black dashed line). The top two targets are occluded between $t = 36$ and $t = 51$. During this interval, no measurements are acquired from these targets. The occlusion region is represented by the green area.

Table 4.4: The simulation parameters for the second scenario.

$\mathbf{X}_{\text{true}}$	Body:	$\begin{bmatrix} 6000 & 0 \\ 0 & 500 \end{bmatrix}$	Bottom Wing:	$\begin{bmatrix} 300 & 500 \\ 500 & 2000 \end{bmatrix}$
	Top Wing:	$\begin{bmatrix} 300 & -500 \\ -500 & 2000 \end{bmatrix}$	Tail:	$\begin{bmatrix} 150 & 0 \\ 0 & 3000 \end{bmatrix}$
$\mathbf{m}_0$	$[-10, 0, 400, 0, 0, -120, 0, 120, -200, 0]^T$			
$\mathbf{P}_0$	$\text{blkdiag}(10\mathbf{I}_4, 100\mathbf{I}_6)$			
$\mathbf{V}_0$	$3000\mathbf{I}_2$			
v_0	30			
$\mathbf{F}_c$	$\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes \mathbf{I}_2$			
$\mathbf{F}_{\mu,\ell}$	$\mathbf{I}_2$			
$\mathbf{F}$	$\text{blkdiag}(\mathbf{F}_c, \mathbf{F}_{\mu,2}, \dots, \mathbf{F}_{\mu,5})$			
$\mathbf{Q}_c$	$\sigma^2 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \otimes \mathbf{I}_2$			
$\mathbf{Q}_{\mu,\ell}$	$\mathbf{I}_2$			
$\mathbf{Q}$	$\text{blkdiag}(\mathbf{Q}_c, \mathbf{Q}_{\mu,2}, \dots, \mathbf{Q}_{\mu,5})$			
$\mathbf{R}$	$10\mathbf{I}_2$			
σ^2	0.1			
T	0.1			

Table 4.5: The RMSE, IOU, GW values and the computation times for the second scenario.

	<i>Sub-obj:</i>	<i>1st</i>	<i>2nd</i>	<i>3rd</i>	<i>4th</i>	Comp. Time Per Time Step[s]
LL	RMSE	6.98	12.46	11.54	14.24	15.97 s
	IOU	0.65	0.65	0.65	0.65	
	GW	26.98	16.15	16.13	18.04	
VPF	RMSE	5.35	11.46	9.86	6.44	351.44 ms
	IOU	0.65	0.70	0.71	0.76	
	GW	32.27	16.76	15.75	11.48	
VB	RMSE	2.58	8.61	7.71	6.73	27.87 ms
	IOU	0.71	0.75	0.75	0.75	
	GW	18.60	11.73	11.02	12.37	

4.5.2 Occlusion Scenario

In this section, we illustrate the capabilities of the algorithm in the presence of an occlusion problem that is frequently encountered in various target tracking applications. For instance, aerial objects might be partly or fully occluded by thick clouds while tracking with a day camera. Many practical systems resort to multiple complementary sensors, such as a thermal and a day camera, to prevent track loss during such occlusions.

In this simulation scenario, we simulated a group of coordinated targets, which contains 5 individuals moving on a straight path in a V-shape formation². In the simulation, 8 measurements per sub-object are generated at every scan. During a certain part of the simulated trajectory, the line of sight of the sensor is partly blocked. It is assumed that no measurements can be obtained from the top two targets of the formation at the corresponding time instances. During the occlusion period, the group of

² Note that, the data generated in this scenario can also be interpreted to originate from a single solid object whose extent is composed of five ellipses.

targets gradually slows down. Then, they increase their velocity incrementally back to their regular pace. A visualization that describes the corresponding scenario is depicted in Fig. 4.5.

We compare the performance of the proposed approach, VB, against the algorithm in [12], referred to as FFK (denoting the initials of the authors' surnames) in the sequel. The VB algorithm incorporates a unified kinematic model, i.e., the sub-objects depend on the common kinematic state as described in Section 4.2. However, FFK treats each sub-object as a different target and tracks them individually without considering the interactions between them. During the occlusion period, FFK relies only on time update equations to estimate the kinematic and extent state of the occluded targets. On the contrary, VB can utilize the measurements that are collected from the visible targets to extract information about occluded targets' state, thanks to the unified kinematics in the model. Consequently, it can provide a much better performance, as shown in Fig. 4.5. Other algorithms, such as [27, 39], which incorporate unified kinematics, can also achieve similar performance in this scenario.

4.5.3 Experiments with Real Data

In this subsection, the performance of the proposed algorithm is demonstrated with real data. The data are extracted from aerial footages of a delta-wing aircraft and a glider.

Our main aims with these experiments are to provide a proof of concept beyond simulations and to present the performance of our approach under measurement model mismatch. Throughout the experiments, each frame of the videos is processed to generate point measurements. Various standard feature extraction algorithms can be used to obtain point measurements from the objects such as Harris corner detector [31], Scale Invariant Feature Transform (SIFT) [50], Speeded Up Robust Features (SURF) [7], Sobel edge detection [38], ORB feature extractor [69] or similar. Here, we present the results obtained using the well-known Sobel edge detection technique [38] and ORB feature extractor [69].

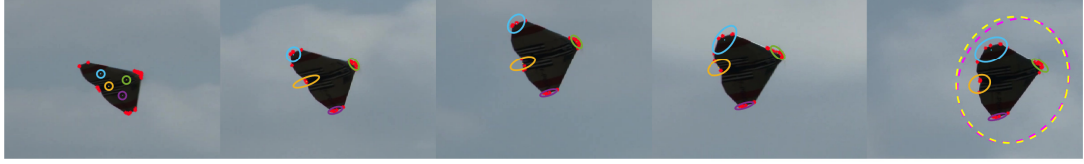
The first real data experiment consists of a delta wing airplane [51]. A representative example of VB algorithm's performance is illustrated in Fig. 4.6a and Fig. 4.6b. Note that the camera zoom is not constant throughout the scenario, which introduces additional challenges for the algorithm.

The second scenario is an air-footage of a glider in motion [36]. The performance of VB algorithm is illustrated in Fig. 4.6c and Fig. 4.6d.

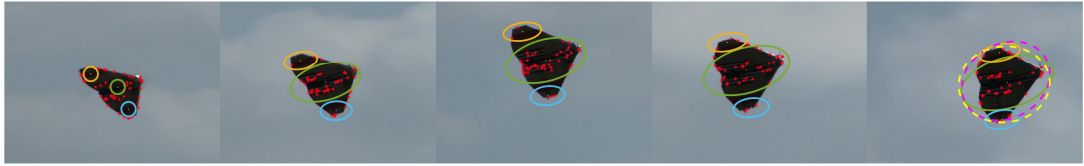
The results indicate that the proposed algorithm can track the measurement generating regions of the objects successfully. As part of the model, a unified kinematics is defined among these regions. This property can be exploited in the case of abrupt changes in the frame characteristics such as background change or occlusions.

In Figure 4.6, we also depict the extent estimates of two different single ellipsoidal target tracking algorithms [18, 90]. For the sake of clarity, only the estimates in the final frames of the video sequences are plotted. Both results demonstrate that modeling the extent with multiple ellipses provides a more accurate representation of the target extent.

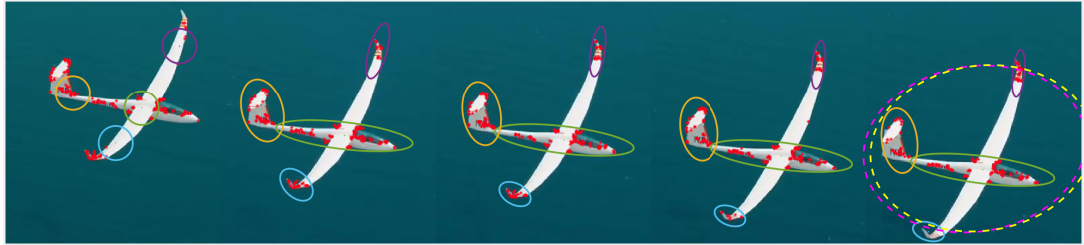
Finally, note that the other multi-ellipsoidal ETT algorithms [27, 39, 45] may also represent the object extent better than the single ellipse tracking approaches in these experiments.



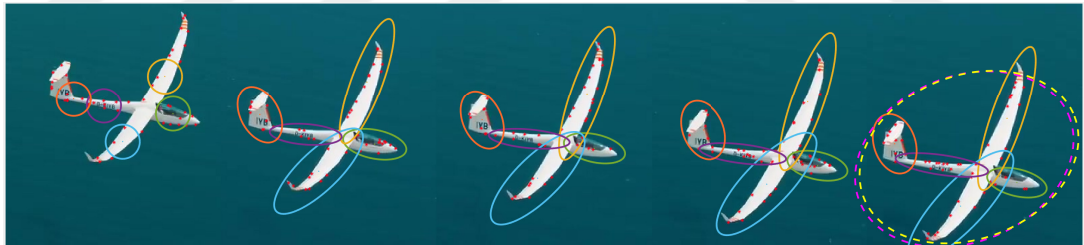
(a) Delta ORB features tracking results.



(b) Delta edge features tracking results.



(c) Glider ORB features tracking results.



(d) Glider edge features tracking results.

Figure 4.6: The results of experiments with real data: (a-b) Delta wing airplane (c-d) Glider. In the last frame of each scenario, the result of the single ellipsoidal target tracking algorithms [90] (magenta dashed line) and [18] (yellow dashed line) are also presented.

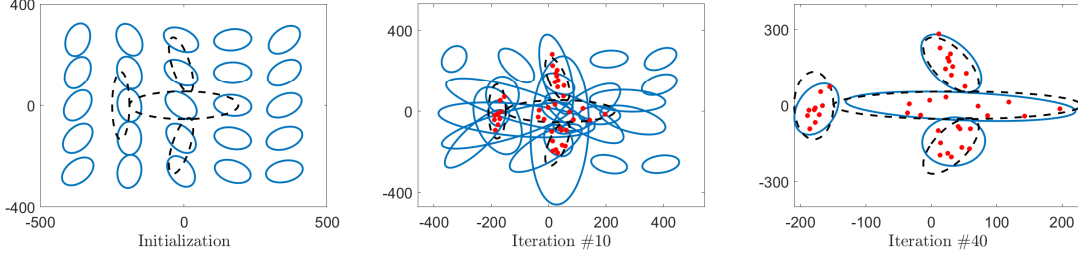


Figure 4.7: An illustration of the results of the variational Gaussian mixture estimation algorithm given in [8, Section 10.2] for selecting the number of sub-objects of the plane-shaped object.

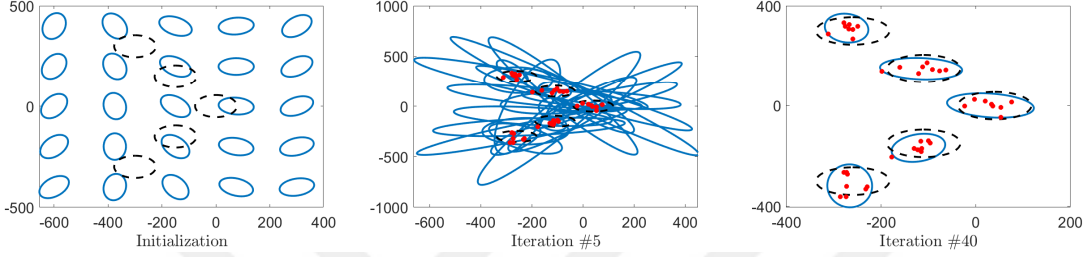


Figure 4.8: An illustration of the results of the variational Gaussian mixture estimation algorithm given in [8, Section 10.2] for selecting the number of sub-objects in the V-shape formation.

4.6 Discussions

4.6.1 The number of sub-objects

Target tracking algorithms mostly aim at resolving the uncertainty in the data in order to make deductions about the existing targets and their states. Consequently, they require some prior information on the structure and parameters of the underlying models. In this respect, we assumed that the number of sub-objects is known and kept as a design parameter which can possibly be determined by offline experiments. In addition, our method is generic in the sense that any Gaussian mixture estimation algorithm that is capable of estimating the number of components at one scan can be employed for pre-processing. Various algorithms can be used for this purpose [15, 28, 57, 79]. As an example, in Figures 4.7 and 4.8, we illustrate the performance of the VB approach given in [8, Section 10.2] on selecting the number of mixture

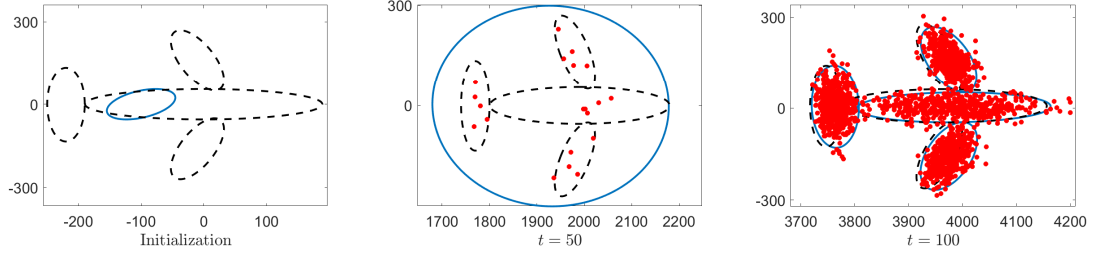


Figure 4.9: An illustration of the results of the variational Gaussian mixture estimation algorithm given in [8, Section 10.2] for selecting the number of sub-objects at a given time during the motion.

components. In the VB approach in [8, Section 10.2] a Gaussian mixture model is employed with priors allowing some of the mixing coefficients to be close to zero. The algorithm is initialized with a mixture of 20 components which are placed on a grid covering the extended object. In every iteration, the components with very small mixing coefficients are removed. As the number of iterations increases, the number of components gradually decreases, settles at a constant value, and a sufficiently good Gaussian mixture model is attained for the plane-shaped object (see Fig. 4.7) and for the targets in V-shape formation (see Fig. 4.8). Another approach to adjust the number of mixture components is to start tracking with a single ellipse, collecting the available measurements up to a given time and initiating the above mentioned algorithm with the aggregated batch of measurements. An example of this scenario is depicted in Fig. 4.9, where measurements are collected until $t = 100$, and a mixture of 20 components is utilized as introduced previously such as in Fig. 4.7 and Fig. 4.8.

4.7 Conclusion and Future Work

In this study, a novel extended target tracking algorithm that is capable of representing a target or a group of targets with multiple ellipses while simultaneously estimating the kinematics is presented. The proposed solution involves approximating the intractable posterior distribution with the variational Bayes method to estimate kinematic and extent states. We demonstrated the performance of the approach in simulations and real data experiments. The results of the experiments show that the proposed

method significantly improves the computation time while achieving better kinematic and extent state estimation performance compared to the existing approaches in the literature.

In this study, mild target maneuvers are handled by using a forgetting factor in the time update. If the severity of the maneuver increases, some performance degradation might occur. In such cases a dedicated algorithm which tracks the orientation of the target together with other kinematic state variables as in [78] would yield a better performance. Similarly, for severe accelerations/decelerations/turn maneuvers, the interacting multiple model framework might be used as in [18].

Finally, in this study, we have assumed that the dynamics of the kinematic state is linear for the sake of simplicity. If the kinematic state evolution is nonlinear, the time update for the kinematic state can be achieved using Gaussian based solutions, such as extended or unscented Kalman filter time updates [70].

CHAPTER 5

EXTENDED TARGET TRACKING AND CLASSIFICATION USING NEURAL NETWORKS

5.1 Introduction

Current advances in intelligent systems, automated vehicles and unmanned aerial vehicles brought the necessity of short-range tracking systems. In contrast to regular long-range counterparts, it is possible to acquire multiple measurements from an object of interest at each instance using short-range sensors. Therefore, they enable us to extract valuable information to a greater extent related to the object contour along with the kinematics of the object, e.g., position, velocity and orientation. In this regard, extended target/object tracking (ETT) algorithms have provided systematic ways to process these measurements to estimate the kinematic state of the object together with its shape. Algorithms presuming simple shape models, such as circle, rectangle, line, are developed in [6, 23, 24]. In this branch of ETT algorithms, the most common approach is to utilize random matrix models, where the target extent is represented by an ellipse [18, 42, 44, 60]. In another line of research, approximate non-parametric models are used to describe the target extent. These methods can simultaneously track and learn various shapes without assuming predefined extents. Random hyper-surface models, as suggested in [4, 5], are examples of this class. More recently, algorithms relying on a Gaussian Process (GP) representation of the unknown target extent have been suggested in [43, 63, 82].

Classification of objects while tracking has been a long-standing problem in the literature. Various algorithms have been proposed to tackle the identification of targets

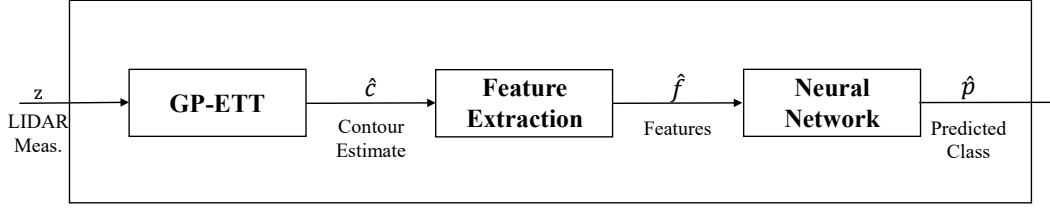


Figure 5.1: Block diagram of the proposed online classification algorithm.

based on their dynamic behavior, motion cues, attributes, fingerprints, etc. [3, 9, 20]. One of the early works on *joint target tracking and classification* (JTC) was presented in [13]. Their method aims to compute the joint target state-class posterior density and allows for cross-coupled feedback between state and class. In [20], authors proposed a particle filter based method, which covers the state and feature space, designed for each class. In [73], authors presented an approach to JTC problem based on belief functions. In another fold of studies, [47] proposed a batch iterative optimization algorithm which minimizes the Bayes risk involving classification and estimation errors. A recursive version of this method is introduced in [49]. Recently, the authors of [11] also tackled the extended object classification problem by relying on a Bayes risk.

In most of the aforementioned methods, there exists a connection between the classification result and the tracking filter, e.g., the result of the classification manipulates the tracking filter. On the contrary, the method proposed in this study considers a rather weak coupling between the tracking and classification tasks, since the output of the tracker is used for classification purposes while there is no established feedback mechanism.

In this study, we consider the problem of online classification of dynamic objects using point cloud data. A block diagram of the proposed method is depicted in Figure 5.1. At each instant, point measurements originated from an object are first processed by GP based extended target tracker (GP-ETT) algorithm which produces estimates of the kinematic state and the contour of the object. After that, the estimated contour is utilized to extract descriptive features of the object shape. These features are then fed to a neural network (NN) model to compute the class probabilities of

the object of interest. Note that we recently addressed the problem of tracking and classification in [77]. The method in [77] essentially relies on a similar architecture which exploits the outputs of GP-ETT for classification; however, it carries out the classification by a Bayesian classifier. Therefore, the contribution of the current study is twofold: retaining the basic structure of our previous study we hereby demonstrate the modularity of the proposed framework, and secondly, we also improve the resulting classification accuracy by utilizing an NN-based classifier.

As a final note, although GP-ETT is employed as the tracker in this study, it can be simply generalized to other ETT algorithms that are capable of generating extent estimates of the objects.

The organization of the chapter is as follows. In Section 5.2, we discuss the feature selection process and the details of the NN model. Subsequently, we briefly introduce the GP-ETT algorithm in Section 5.3 to provide an insight into the contour representation that we rely on for classification. This is followed by the demonstration of the classification performance via simulation results in Section 5.4. Finally, we conclude our work in Section 5.5.

5.2 Object Classification

In this study, we aim at achieving object classification by using contour estimates produced by an extended object tracker. With this purpose, the estimated contour is to be transformed into some features to acquire a descriptive representation of the object shape. Thereafter, an NN-based classifier will perform the classification task regarding these features. In this section, the details of the selected features and the classifier architecture are discussed.

5.2.1 Feature Selection for Contour Representation

In literature, studies on shape classification mostly rely on two different interpretations of the object shape. These interpretations are embodied by either region-

based descriptors [41] or contour-based descriptors [46], [10]. Considering substantial amount of empirical evidence, an object can be described as a combination of a set of regions or may be a single body. These regions might include some holes inside. Region-based descriptors make use of all information constituting the shape including the holes or several disjoint regions. On the other hand, contour-based descriptors consider the characteristic shape features extracted from the contour of an object while ignoring what is inside the contour. In this study, we naturally direct our attention to the utilization of the contour-based descriptors since we make use of contour estimates obtained by the tracking algorithm and none of the shapes consist of any holes or disjoint members.

There are various contour descriptors proposed in the literature. The most common ones are the *Fourier Descriptors* (FD), [66, 83], and *Curvature Scale Space* (CSS), [54, 55].

FD consist of the Fourier coefficients of 2-D shapes which can be represented in terms of different shape signatures, such as the contour coordinates expressed as complex numbers or the radial distance between the contour and the object center, [83]. FD are robust to rotation and affine transformation, while being efficient in terms of computation. CSS, on the other hand, considers the inflection points as descriptors which represent the location of change in the direction of the curvature. The contour is convolved with multiple Gaussian kernels with different standard deviations, also called as width, to smoothen the contour. The inflection points are calculated for different Gaussian kernels and used as descriptors. One of the main advantages of CSS is that it is noise invariant since the characteristic inflection points remain available after filtering with large width kernels, while other points resolve.

In addition to these prominent descriptors, there are also numerous geometric features with various levels of complexity, [86], [87], [75]. These features are basically employed to map object contours into some descriptive representations. Selecting a proper set of features among numerous alternatives is of paramount importance for classification performance. In this regard, we restrict our scope to the simple geometric features to be able to form a basis for computationally efficient and fast operation.

The following set of six features are chosen to be respected by the classifier: *elongation* [87], *rectangularity* [86], *circularity* [86], *solidity* [86], *compactness* [75], and *area*. Note that this particular group of features is not hand-crafted to optimize the resulting classification performance. Instead, it is empirically observed to be sufficient for the proof of concept. However, for a specific application, it can be selected to account for any prior knowledge or can be learned in an automated fashion by employing various tools. such as convolutional neural networks. autoencoders.

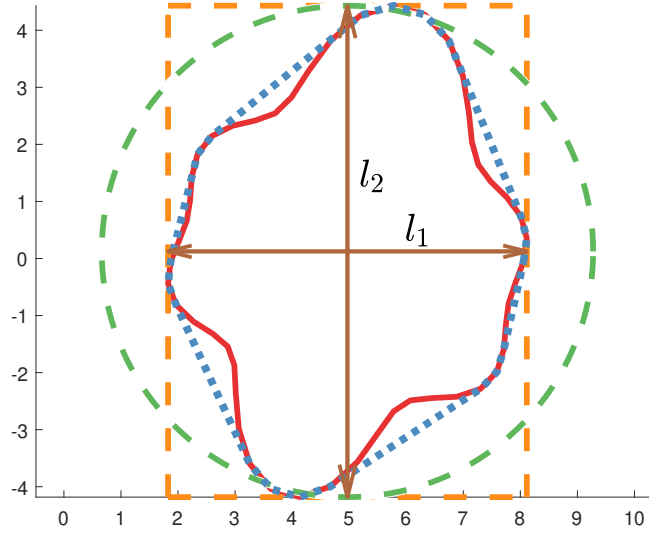


Figure 5.2: Visualization of the parameters used in the feature extraction process. The solid red curve represents the contour of the object. The minimum bounding circle and rectangle are plotted in green and yellow, respectively. Major and minor axes of the minimum bounding rectangle are denoted by l_1 and l_2 . The blue dashed line indicates the convex hull of the object.

Elongation is uniquely defined as the ratio of the major and minor axes of the minimum bounding rectangle (which are denoted as l_1 and l_2 in Fig. 5.2). Rectangularity is the ratio of the object area to the area of the minimum size rectangle that encloses the contour. In Fig. 5.2, rectangularity can be computed as the object area to the area encapsulated by the orange contour. On the other hand, circularity is related to the ratio of the object area to the area of the minimum confining circle, and it is basically used to measure the similarity of the object of interest to a circle. The minimum bounding circle is illustrated by the green line in Fig. 5.2, and the circularity is calculated as the ratio of the object area to the area enclosed by the green contour. Solidity

is an indicator of the shape being convex or concave. It is defined as the ratio of the object area to the area of the convex hull. The convex hull of the shape is represented by the blue line in Fig. 5.2. Lastly, compactness measures the contour complexity versus the enclosed area. The value of compactness increases with increasing shape complexity. The expression to calculate the compactness is given as [75]

$$C = 1 - \frac{4\pi a}{p^2}, \quad (5.1)$$

where a and p are the area and the perimeter of the shape, respectively.

The computed features of the object contour are then passed to the NN-based classifier. The classifier architecture is revealed in the following subsection.

5.2.2 Classifier Architecture

In this study, we realize object classification by a naively deep feedforward NN. A typical NN model is basically a collection of special processing units, called neurons, which are grouped into different layers, such as input, hidden and output layers. Specifically, in a feedforward NN, there exist weighted connections between these layers while there is no connection between neurons within the same layer. A simple feedforward NN consisting of one input, one hidden and one output layer is depicted in Fig. 5.3.

In vector form, the outputs of the neurons in the hidden layer can be computed as follows.

$$\mathbf{a} = \mathbf{g}(\mathbf{W}_i^T \mathbf{x} + \mathbf{b}_i) \quad (5.2)$$

$\mathbf{a} \triangleq [a_1 \dots a_m]^T$ represents the output vector of the hidden layer; $\mathbf{x} \triangleq [x^{(1)} \dots x^{(N_i)}]^T$ indicates the input vector of the network; $\mathbf{W}_i$ is the input weight matrix; $\mathbf{b}_i \triangleq [b_1 \dots b_m]^T$ denotes the bias vector; $\mathbf{g}(\cdot)$ is the activation function. Note that if the NN model has multiple hidden layers, the output of k^{th} hidden layer can easily be calculated similarly via replacing $\mathbf{x}$ in (5.2) by the output of the $(k - 1)^{th}$ layer.

Besides, the output of the network is obtained by

$$\hat{\mathbf{y}} = \mathbf{W}_o^T \mathbf{a} + \mathbf{b}_o \quad (5.3)$$

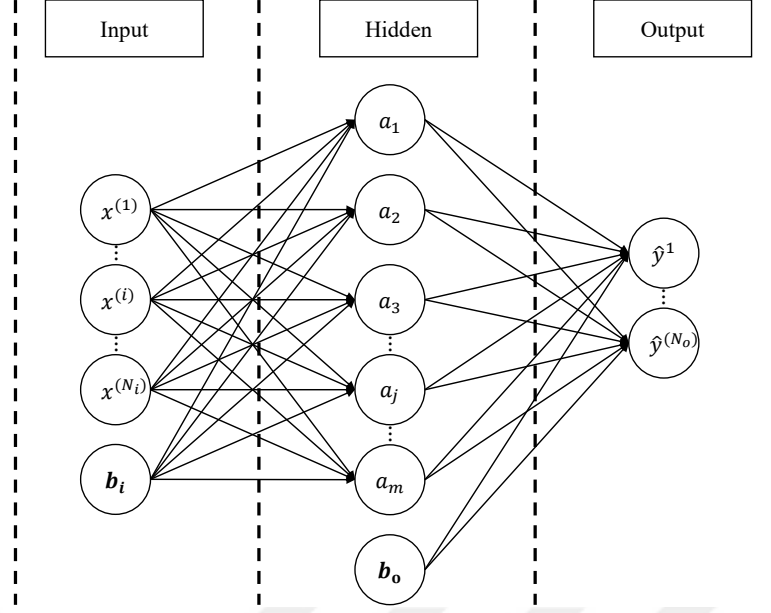


Figure 5.3: A simple neural network including one input layer, one hidden layer and one output layer.

where $\hat{\mathbf{y}} \triangleq [\hat{y}_1 \dots \hat{y}_{N_o}]^T$ is the output vector; $\mathbf{W}_o$ is the output weight matrix and $\mathbf{b}_o$ is the bias vector.

The standard loss function used in the training procedure to optimize the parameters of an NN is given in (5.4).

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N (y^{(i)} - f_{\theta}(x^{(i)}))^2 \quad (5.4)$$

There are N training points and $y^{(i)}$ denotes the ground truth for the i^{th} input $x^{(i)}$. The function $f_{\theta}(x^{(i)})$ is the output of the NN for the corresponding input, and it is parametrized by θ which comprises of the weights and the biases of the model.

In this study, we construct a naively deep NN consisting of 2 hidden layers with respectively 16 and 8 neurons. This specific architecture was empirically observed to be sufficient regarding the classification performance obtained for training and validation data sets. The activation functions are selected as *hyperbolic tangent sigmoid* function in the hidden layers and *softmax* function in the output layer.

5.3 GP-ETT

First introduced in [82], GP-ETT is an effective way of tracking dynamic objects with unknown shapes. It is able to jointly estimate the kinematics and the extent of the object by using point measurements. The suggested NN-based object classification scheme basically processes the contour estimates produced by this algorithm.

In the formulation of GP-ETT, the contour of an object is represented by a radial function $r = f(\theta)$ which is to be modeled by a GP. The output of the radial function is the distance between the center and the contour at the specified polar angle. A typical shape described by radial function is shown in Fig. 5.4. Notice that the formulation implicitly assumes that the shape of the object is *star-convex*¹.

A noisy observation of the object contour represented by the radial function can be described as

$$\mathbf{z}_{k,l} = \mathbf{x}_k^c + \mathbf{p}(\theta_{k,l})f(\theta_{k,l}) + \mathbf{e}_{k,l}, \quad (5.5)$$

where $\mathbf{x}_k^c$ is the center position of the target at time instant k , $\{\mathbf{z}_{k,l}\}_{l=1}^{n_k}$ indicate the measurements collected at time k , $\{\theta_{k,l}\}_{l=1}^{n_k}$ are the polar angles of the source points on the contour that originate the corresponding measurements, $\mathbf{e}_{k,l} \sim \mathcal{N}(0, \mathbf{R})$ denote i.i.d. Gaussian noise with zero mean and covariance $\mathbf{R}$, and $\mathbf{p}(\theta_{k,l})$ is an orientation vector defined as $\mathbf{p}(\theta_{k,l}) \triangleq \begin{bmatrix} \cos(\theta_{k,l}) \\ \sin(\theta_{k,l}) \end{bmatrix}^T$.

The core idea of GP-ETT is to facilitate modeling of the unknown radial function $f(\cdot)$ in (5.5) by a GP, i.e., $f(\theta) \sim \mathcal{GP}(\mu(\theta), k(\theta, \theta'))$ where $\mu(\cdot)$ is the mean function and $k(\cdot, \cdot)$ is the covariance function. By doing so, one can establish a probabilistic representation of the extent whose inherent spatial characteristics are conveniently encoded by the covariance function. In this study, the GP is specified with the zero mean function and the following covariance function, which is obtained by modifying the squared exponential kernel.

$$k(\theta, \theta') = \sigma_f^2 e^{-\frac{2\sin^2\left(\frac{|\theta - \theta'|}{2}\right)}{l^2}} + \sigma_r^2 \quad (5.6)$$

¹ A set $\mathcal{S}(x)$ is called star-convex if each line segment from the center to any point is fully contained in $\mathcal{S}(x)$, where x denotes the position of a point.

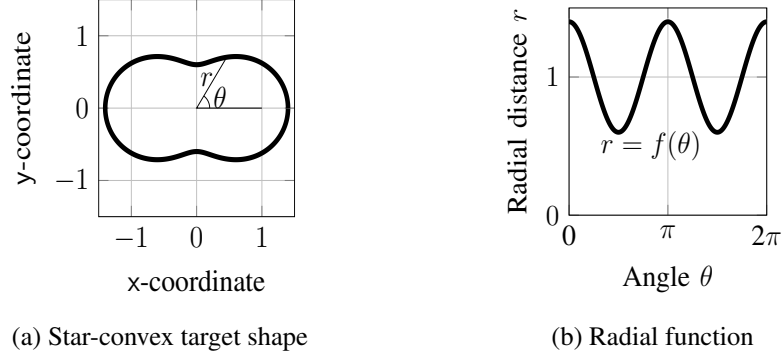


Figure 5.4: An example star-convex contour described by radial function $r = f(\theta)$, [82].

σ_f is the prior variance, l is the scaled length and σ_r essentially accounts for the uncertainty in the mean. The covariance function is illustrated in Fig. 5.5. Being a stationary covariance function which depends on the relative position of the inputs, the function is plotted against the difference between the input angles, i.e., $\theta - \theta'$. The figure reveals that the covariance function ensures perfect correlation between $f(\theta)$ and $f(\theta + 2\pi)$, i.e., $\rho(f(\theta), f(\theta')) = 1$, since they basically correspond to the same point on the contour. Additionally, the correlation between $f(\theta)$ and $f(\theta')$ decreases as the angle between them increases. This characteristic essentially accounts for the fact that radii at closer angles naturally tend to be more interrelated than the farther sections of the contour.

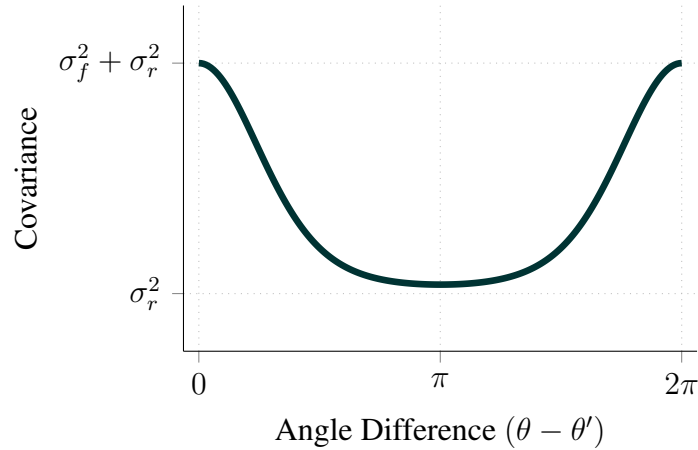


Figure 5.5: The covariance function of the GP as defined in (5.6).

The standard GP regression requires batch processing of the measurements hence it

does not apply to the online tracking and classification application due to its computational complexity and sequentially available measurements. To overcome these issues, an approximation of the standard regression was suggested to compute the posterior recursively in [82]. In particular, this method summarizes the original GP model at a finite number of basis inputs $\mathbf{u} = [u_1 \ \dots \ u_N]^T$, thus the posterior distribution can simply be computed by a Kalman filter regarding the following state space model.

$$\mathbf{c}_{k+1} = F\mathbf{c}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}), \quad (5.7a)$$

$$z_k = H(u_k)\mathbf{c}_k + \mathbf{e}, \quad \mathbf{e} \sim \mathcal{N}(0, \mathbf{R}(u_k)), \quad (5.7b)$$

$$\mathbf{c}_0 \sim \mathcal{N}(0, \mathbf{P}_0), \quad (5.7c)$$

where $\mathbf{c}_k = [f(u_1) \ \dots \ f(u_N)]^T$ is the radial function values at time k , and z_k indicates a single point measurement. Further details of the model can be found in [82].

To be able jointly estimate the target dynamics and the contour, a unified state space model is constructed which relies on the state vector, $\mathbf{x}_k \triangleq [\bar{\mathbf{x}}_k \ \mathbf{c}_k]$, where $\bar{\mathbf{x}}_k$ denotes the object dynamics including position, velocity and orientation. GP-ETT algorithm is simply implemented by an extended Kalman filter (EKF) considering this state vector. The EKF infers the joint posterior of the state, $p(\mathbf{x}_k | \mathbf{z}_{1:k})$, and the updated mean of the contour $\hat{c}$ is delivered to the classification scheme as the contour estimate.

As GP-ETT recursively updates the posterior of the state, it thereby refines the contour estimate due to the accumulation of information in time. In contrast, instant point measurements can only delineate the shape partially, and they get sparse due to occlusions or increased distance between the object and the sensor. Consequently, GP-ETT provides a more reliable basis to perform classification compared to instant measurements.

5.4 Simulations and Results

In this section, the performance of the proposed algorithm is demonstrated through some simulation experiments. Throughout simulations, objects from the following shape classes are taken into consideration: circle, triangle, rectangle and plus. To form a labeled data set, first we simulate dynamic scenarios of random sized objects from each class. The initial orientation of each object is randomly selected from a uniform distribution between 0 and 2π . At each instant of the simulations, point measurements are originated from random points on the object contour with an additive Gaussian noise whose standard deviation is set to 0.02. Subsequently, these point measurements are processed by the GP-ETT to obtain corresponding contour estimates. Contour estimates are maintained at fifty basis angles which are evenly spaced between 0- 2π . Some typical examples of the contour estimates from each class are exhibited in Fig. 5.6. The resulting data set consists of 10000 contour estimates per each shape class, and it is divided into training and test sets by the ratios of 80% and 20%, respectively. The training set is further split into two by the same ratios, 80% and 20%, to attain training and validation sets.

To be able to assess the performance of the proposed algorithm in a comparative manner, we consider another classification method from our previous work, [77]. This baseline algorithm is based on a Bayesian classification scheme and is denoted as ‘BC’. Different from the proposed method, BC makes use of both the contour estimate and the associated covariance matrix, which essentially carries the local uncertainty information along the contour estimate. In particular, BC applies Unscented Transform (UT) to the outputs of GP-ETT to adopt a probabilistic representation in the feature space. Subsequently, the classification is achieved by the following Bayesian classifier which regards this probabilistic description of the feature.

$$Pr(r = i|f) = \frac{p(f|r = i)p(r = i)}{\sum_{j=1}^M p(f|r = j)p(r = j)} \quad (5.8)$$

where f is the feature vector; $r \in \{1, \dots, M\}$ is the class index, and $p(f|r = j)$ is the likelihood of observing the feature f from an object in class- j . The posterior probability, $Pr(r = i|f)$, can simply be computed for each class once the sufficient

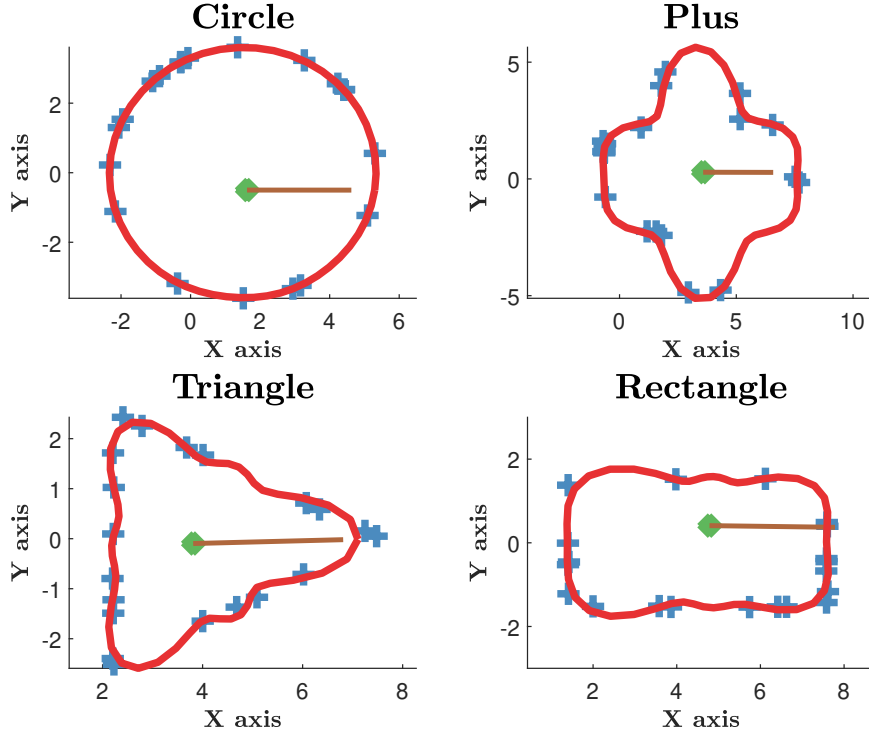


Figure 5.6: Typical contour estimates produced by GP-ETT. (Contour estimates are plotted in blue while red crosses represent point measurements.)

statistics of the class distributions are available. To this end, the feature distribution of each class is approximated to be Gaussian, i.e., $p(f|r = i) \approx \mathcal{N}(f; \mu_i, \Sigma_i)$, and the mean, μ_i , and the covariance matrix, Σ_i , are determined by means of a supervised learning scheme using the labeled training data. For further details, see [77].

Additionally, we also implemented a variant of the proposed method which employs an NN having the same architecture; however, this time the NN is trained and tested directly on the contour estimates rather than the features extracted from these estimates. The original method is denoted as ‘NN-feature’ and ‘NN-contour’ stands for the variant.

All of the algorithms were implemented in MATLAB 2018a; in particular, we used Deep Learning Toolbox for NN-feature and NN-contour. Since the NN model is a rather shallow one, we optimized the network parameters by Levenberg-Marquardt method, which provides a fast but computationally expensive optimization scheme combining the gradient descent algorithm with Gauss-Newton method. The training

procedure is stopped when the validation accuracy does not increase for the following 20 consecutive epochs. The loss function values of NN-feature at each epoch are demonstrated for both training and validation sets in Fig. 5.7. The training and test of the models were performed on a computer with an Intel Core i7-6700HQ CPU

v

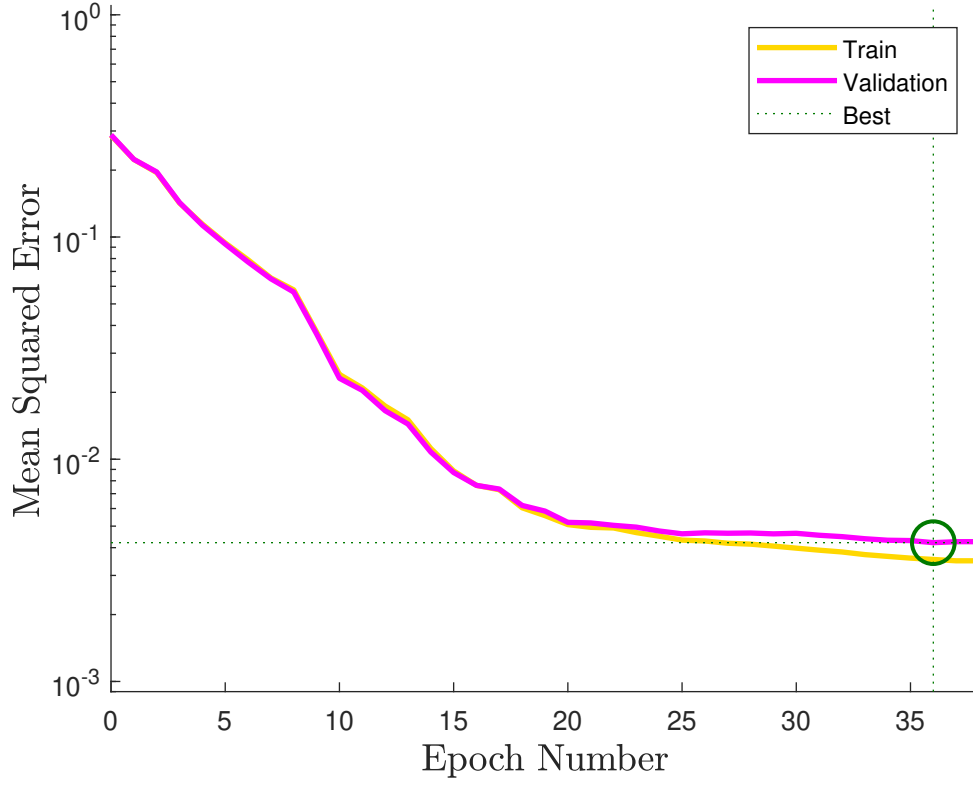


Figure 5.7: The performance curves of the proposed model (NN-feature). The network parameters are set to be the values satisfying the best performance in the validation set.

The results obtained by the methods are presented in Table 5.1. NN-feature is observed to outperform the other algorithms in the accuracy rate. Specifically, the difference in the accuracy rates of NN-feature and BC is mainly due to the test scenarios including irregular contour estimates generated due to insufficient sampling of the objects. Some typical classification outputs produced at these instants are depicted in Table 5.2. NN-feature can handle these challenging scenarios more robustly. Even though NN-contour has a higher accuracy rate compared to BC, the execution time per object is significantly higher than the others. Particularly, NN-contour operates more

slowly compared to NN-feature because the former processes fifty contour points while the latter one considers only six features. This characteristic might render NN-contour inconvenient for object classification in real time.

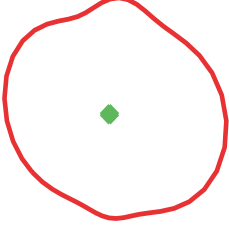
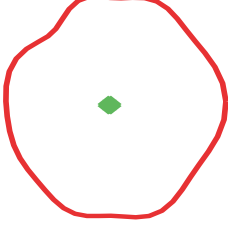
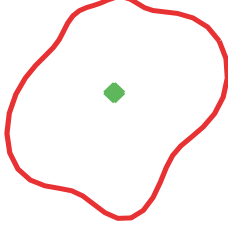
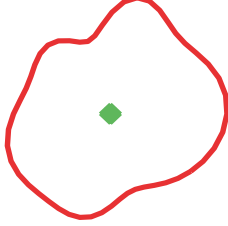
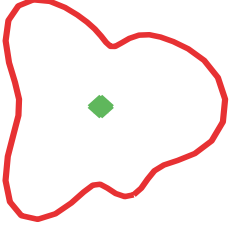
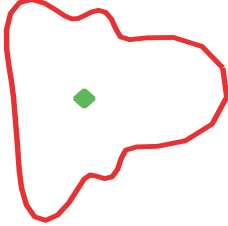
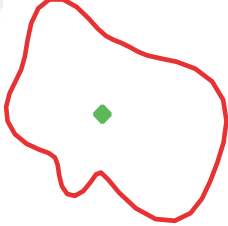
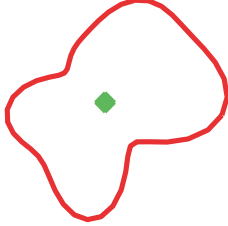
Table 5.1: Classification results on the synthetic data set in terms of accuracy rate and the execution times for each shape.

Classifier	Accuracy Rate	Execution Time (ms/object)
BC	0.94	0.57
NN-feature	0.99	1.80
NN-contour	0.97	2.55

A common problem encountered during the training phase of an NN is over-fitting to the training data. This leads to degradation in the generalization capabilities of the NN to other data sets. With this in mind, we briefly examined the effect of $\mathcal{L}_2$ regularization on the performance of NN-feature and NN-contour. For both of the NN-based algorithms, regularization did not improve the accuracy rate of the corresponding model.

It can be concluded that utilization of an NN in the classifier yields more robust performance especially for irregular contour estimates, which is frequently encountered in real-world scenarios when the tracked object is occluded by its surroundings. However, the simplicity of the Bayesian classifier shows its strength at the execution time by being approximately three times faster than the proposed method.

Table 5.2: Some of the results that NN shows superior performance against Bayesian approach.

 <i>Ground Truth:</i> Circle <i>Bayesian:</i> Plus <i>NN:</i> Circle	 <i>Ground Truth:</i> Plus <i>Bayesian:</i> Circle <i>NN:</i> Plus	 <i>Ground Truth:</i> Plus <i>Bayesian:</i> Rectangle <i>NN:</i> Plus	 <i>Ground Truth:</i> Plus <i>Bayesian:</i> Rectangle <i>NN:</i> Plus
 <i>Ground Truth:</i> Triangle <i>Bayesian:</i> Rectangle <i>NN:</i> Triangle	 <i>Ground Truth:</i> Triangle <i>Bayesian:</i> Plus <i>NN:</i> Triangle	 <i>Ground Truth:</i> Rectangle <i>Bayesian:</i> Plus <i>NN:</i> Rectangle	 <i>Ground Truth:</i> Rectangle <i>Bayesian:</i> Triangle <i>NN:</i> Rectangle

5.5 Conclusion and Future Work

Classifying objects based on their extent estimates is still at very early ages in the ETT literature. In this study, we propose to combine a well-known deep learning algorithm with a GP based extended target tracker to classify the type of dynamic objects. To this end, various shape features and different structures of NNs are examined. The performance of the suggested algorithm is comparatively demonstrated against a Bayesian classifier. NN-based classifier shows superior performance compared to the Bayesian classifier. As future work, the algorithm will be tested on real data sets considering several applications such as identification of biological cells and annotating agents in urban driving environment.



CHAPTER 6

CONCLUSION AND FUTURE WORK

Target tracking is an emerging topic with the recent developments in sensor technology and autonomous vehicles. The earlier algorithms, which are capable of representing the target with a single point, gave their place to more comprehensive solutions with which one can also gather information such as the shape of the target and heading angle. Researchers proposed various solutions to estimate the extent of the target with respect to the measurements. Some simple approaches denote the target's shape with a predefined simple shape such as a circle, rectangle, and stick. On the other hand, more complex solutions include representing the extent with a random hyper-surface model, an ellipsoid using a random matrix model, or a non-parametric Gaussian process model.

In the scope of this thesis, we have proposed two novel extended target tracking algorithms on random matrix models. In our first work, we aimed to solve the computation of the orientation problem while simultaneously estimating the extent of the object with a novel model and inference. The proposed solution utilizes appropriate priors, does not rely on multiplicative noise terms or pseudo-measurements, and is easy to implement. The second contribution incorporates a novel approach representing a target's extent or a group of targets with multiple ellipsoidal shapes. The suggested algorithm has low computational complexity and does not require any clustering, partitioning, mixture reduction, and merging steps. We have utilized the well-known variational Bayes inference technique in both studies to achieve the desired outcomes. In the result sections of the studies, we showed the performances and capabilities of the methods through simulations and real data experiments. Ex-

perimental results demonstrate that the proposed solutions significantly improve the tracking performance and the accuracy in estimating the extent of the target.

In another fold of study, we have presented that the output of the extended target tracking algorithms could be used as a base to further research purposes such as the classification of the targets. We have modeled a deep neural network on top of a Gaussian process-based extended target tracker to classify targets based on their extent information. We compared the performance of our solution against a Bayesian classifier.

In the future, we intend to combine our two extended target tracking algorithms to obtain an approach that can approximate the orientation while representing the extent with multiple ellipses. Additionally, we aim to leverage our work to incorporate the estimation of the orientation's derivative. On the classification side, we aspire to elevate our algorithm into 3D real-life applications. On a broader vision, we hope that our research will serve as a base for other works in the extended target tracking community.

REFERENCES

- [1] W. Aftab, A. De Freitas, M. Arvaneh, and L. Mihaylova. A Gaussian process approach for extended object tracking with random shapes and for dealing with intractable likelihoods. In *2017 22nd International Conference on Digital Signal Processing (DSP)*, pages 1–5. IEEE, 2017.
- [2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
- [3] Y. Bar-Shalom, P. K. Willett, and X. Tian. *Tracking and data fusion*. YBS publishing, 2011.
- [4] M. Baum and U. D. Hanebeck. Random hypersurface models for extended object tracking. In *International Symposium on Signal Processing and Information Technology*, pages 178–183. IEEE, 2009.
- [5] M. Baum and U. D. Hanebeck. Shape tracking of extended objects and group targets with star-convex RHMs. In *14th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2011.
- [6] M. Baum, V. Klumpp, and U. D. Hanebeck. A novel Bayesian method for fitting a circle to noisy points. In *13th International Conference on Information Fusion (FUSION)*, pages 1–6. IEEE, 2010.
- [7] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.
- [8] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

- [9] S. Blackman and R. Popoli. *Design and analysis of modern tracking systems*. Artech house, 1999.
- [10] M. Bober. MPEG-7 visual shape descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):716–719, 2001.
- [11] W. Cao, J. Lan, and X. R. Li. Extended object tracking and classification using radar and ESM sensor data. *IEEE Signal Processing Letters*, 25(1):90–94, 2018.
- [12] D. Carevic and S. Davey. Two algorithms for modeling and tracking of dynamic time-frequency spectra. *IEEE Transactions on Signal Processing*, 64(22):6030–6045, 2016.
- [13] S. Challa and G. W. Pulford. Joint target tracking and classification using radar and ESM sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 37(3):1039–1055, 2001.
- [14] W. Chen, D. Ding, H. Dong, and G. Wei. Distributed resilient filtering for power systems subject to denial-of-service attacks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(8):1688–1697, 2019.
- [15] A. Corduneanu and C. M. Bishop. Variational Bayesian model selection for mixture distributions. In *Artificial Intelligence and Statistics*, volume 2001, pages 27–34. Morgan Kaufmann Waltham, MA, 2001.
- [16] D. Ding, Z. Wang, and Q.-L. Han. A set-membership approach to event-triggered filtering for general nonlinear systems over sensor networks. *IEEE Transactions on Automatic Control*, 65(4):1792–1799, 2019.
- [17] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.
- [18] M. Feldmann, D. Franken, and W. Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4):1409–1420, 2011.

- [19] C. R. Givens and R. M. Shortt. A class of Wasserstein metrics for probability distributions. *The Michigan Mathematical Journal*, 31(2):231–240, 1984.
- [20] N. J. Gordon, S. Maskell, and T. Kirubarajan. Efficient particle filters for joint tracking and classification. In *Signal and Data Processing of Small Targets*, volume 4728, pages 439–450. International Society for Optics and Photonics, 2002.
- [21] K. Granström. An extended target tracking model with multiple random matrices and unified kinematics. *CoRR*, abs/1406.2135, 2014.
- [22] K. Granstrom, M. Baum, and S. Reuter. Extended object tracking: Introduction, overview and applications. *arXiv preprint arXiv:1604.00970*, 2016.
- [23] K. Granström and C. Lundquist. On the use of multiple measurement models for extended target tracking. In *16th International Conference on Information Fusion (FUSION)*, pages 1534–1541. IEEE, 2013.
- [24] K. Granström, C. Lundquist, and U. Orguner. Tracking rectangular and elliptical extended targets using laser measurements. In *14th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2011.
- [25] K. Granström and U. Orguner. New prediction for extended targets with random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 50(2):1577–1789, Apr. 2014.
- [26] K. Granström, S. Renter, M. Fatemi, and L. Svensson. Pedestrian tracking using velodyne data—Stochastic optimization for extended object tracking. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 39–46. IEEE, 2017.
- [27] K. Granström, P. Willett, and Y. Bar-Shalom. An extended target tracking model with multiple random matrices and unified kinematics. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1007–1014. IEEE, 2015.
- [28] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

- [29] L. Guerlin, B. Pannetier, M. Rombaut, and M. Derome. Study on group target tracking to counter swarms of drones. In *Signal Processing, Sensor/Information Fusion, and Target Recognition XXIX*, volume 11423, page 1142304. International Society for Optics and Photonics, 2020.
- [30] A. K. Gupta and D. K. Nagar. *Matrix variate distributions*, volume 104. CRC Press, 2018.
- [31] C. G. Harris, M. Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.
- [32] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. Eslami, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.
- [33] T. Hirscher, A. Scheel, S. Reuter, and K. Dietmayer. Multiple extended object tracking using Gaussian processes. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 868–875. IEEE, 2016.
- [34] Q. Hu, H. Ji, and Y. Zhang. Tracking of maneuvering non-ellipsoidal extended target with varying number of sub-objects. *Mechanical Systems and Signal Processing*, 99:262–284, 2018.
- [35] Y. Huang, Y. Zhang, Z. Wu, N. Li, and J. Chambers. A novel adaptive Kalman filter with inaccurate process and measurement noise covariance matrices. *IEEE Transactions on Automatic Control*, 63(2):594–601, 2017.
- [36] Joyplanes RC. “How can can gliders fly without propulsion | The most complete explanation” Aug. 23, 2019. [YouTube video, used with permission].
- [37] R. E. Kalman. A new approach to linear filtering and prediction problems. 1960.
- [38] N. Kanopoulos, N. Vasanthavada, and R. L. Baker. Design of an image edge detection filter using the Sobel operator. *IEEE Journal of Solid-State Circuits*, 23(2):358–367, 1988.

- [39] S. F. Kara and E. Özkan. Multi-ellipsoidal extended target tracking using sequential Monte Carlo. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2018.
- [40] S. F. Kara and E. Özkan. Multi-ellipsoidal extended target tracking using sequential Monte Carlo. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2018.
- [41] H.-K. Kim and J.-D. Kim. Region-based shape descriptor invariant to rotation, scale and translation. *Signal Processing: Image Communication*, 16(1-2):87–93, 2000.
- [42] J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, 2008.
- [43] M. Kumru and E. Özkan. 3D extended object tracking using recursive Gaussian processes. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2018.
- [44] J. Lan and X. R. Li. Tracking of extended object or target group using random matrix—Part I: New model and approach. In *15th International Conference on Information Fusion (FUSION)*, pages 2177–2184. IEEE, 2012.
- [45] J. Lan and X. R. Li. Tracking of maneuvering non-ellipsoidal extended object or target group using random matrix. *IEEE Transactions on Signal Processing*, 62(9):2450–2463, 2014.
- [46] L. J. Latecki, R. Lakamper, and T. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 424–429. IEEE, 2000.
- [47] X. R. Li. Optimal Bayes joint decision and estimation. In *10th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2007.
- [48] S. Linnainmaa. The representation of the cumulative rounding error of an al-

gorithm as a Taylor expansion of the local rounding errors. *Master's Thesis (in Finnish)*, Univ. Helsinki, pages 6–7, 1970.

- [49] Y. Liu and X. R. Li. Recursive joint decision and estimation based on generalized Bayes risk. In *14th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2011.
- [50] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [51] Marc Talloen. “Verhees Delta Flight and LANDING of that little Plane Sanicole 2012 part II.” Dec. 30, 2018. [YouTube video, used with permission].
- [52] H. Mendoza, A. Klein, M. Feurer, J. T. Springenberg, and F. Hutter. Towards automatically-tuned neural networks. In *Workshop on Automatic Machine Learning*, pages 58–65. PMLR, 2016.
- [53] M. Michaelis, P. Berthold, D. Meissner, and H.-J. Wuensche. Heterogeneous multi-sensor fusion for extended objects in automotive scenarios using Gaussian processes and a GMPHD-filter. In *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6. IEEE, 2017.
- [54] F. Mokhtarian and A. Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):34–43, 1986.
- [55] F. Mokhtarian and A. K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):789–805, 1992.
- [56] K. P. Murphy. Conjugate Bayesian analysis of the Gaussian distribution. *def*, 1(2 σ 2):16, 2007.
- [57] R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

- [58] H. Nurminen, T. Ardeshiri, R. Piche, and F. Gustafsson. Robust inference for state-space models with skewed measurement noise. *IEEE Signal Processing Letters*, 22(11):1898–1902, 2015.
- [59] H. Nurminen, T. Ardeshiri, R. Piché, and F. Gustafsson. Skew- t filter and smoother with improved covariance matrix approximation. *IEEE Transactions on Signal Processing*, 66(21):5618–5633, 2018.
- [60] U. Orguner. A variational measurement update for extended target tracking with random matrices. *IEEE Transactions on Signal Processing*, 60(7):3827–3834, 2012.
- [61] E. Özkan, I. Ozbek, and M. Demirekler. Dynamic speech spectrum representation and tracking variable number of vocal tract resonance frequencies with time-varying Dirichlet process mixture models. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(8):1518–1532, Nov 2009.
- [62] E. Özkan, V. Šmídl, S. Saha, C. Lundquist, and F. Gustafsson. Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters. *Automatica*, 49(6):1566–1575, 2013.
- [63] E. Özkan, N. Wahlström, and S. J. Godsill. Rao-Blackwellised particle filter for star-convex extended target tracking models. In *19th International Conference on Information Fusion (FUSION)*, pages 1193–1199. IEEE, 2016.
- [64] A. Papoulis and S. U. Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.
- [65] G. Parisi and R. Shankar. Statistical field theory. *Physics Today*, 41(12):110, 1988.
- [66] E. Persoon and K.-S. Fu. Shape discrimination using Fourier descriptors. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(3):170–179, 1977.
- [67] X. Qing, F. Yang, and X. Wang. Extended set-membership filter for power system dynamic state estimation. *Electric power systems research*, 99:56–63, 2013.

- [68] C. E. Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.
- [69] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *International Conference on Computer Vision*, pages 2564–2571. IEEE, 2011.
- [70] S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
- [71] S. Sarkka and A. Nummenmaa. Recursive noise adaptive Kalman filtering by variational Bayesian approximations. *IEEE Transactions on Automatic control*, 54(3):596–600, 2009.
- [72] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström. Mono-camera 3D multi-object tracking using deep learning detections and pmbm filtering. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 433–440. IEEE, 2018.
- [73] P. Smets and B. Ristic. Kalman filter and joint tracking and classification based on belief functions in the TBM framework. *Information Fusion*, 8(1):16–27, 2007.
- [74] V. Smidl and A. Quinn. Variational Bayesian filtering. *IEEE Transactions on Signal Processing*, 56(10):5020–5030, 2008.
- [75] Y. Tao, S.-C. B. Lo, M. T. Freedman, and J. Xuan. A preliminary study of content-based mammographic masses retrieval. In *Medical Imaging 2007: Computer-Aided Diagnosis*, volume 6514. International Society for Optics and Photonics, 2007.
- [76] B. Tuncer, M. Kumru, and E. Özkan. Extended target tracking and classification using neural networks. In *2019 22th International Conference on Information Fusion (FUSION)*, pages 1–7. IEEE, 2019.
- [77] B. Tuncer, M. Kumru, E. Özkan, and A. A. Alatan. Extended object tracking

- and shape classification. In *21st International Conference on Information Fusion (FUSION)*, pages 1–5. IEEE, 2018.
- [78] B. Tuncer and E. Özkan. Random matrix based extended target tracking with orientation: A new model and inference. *IEEE Transactions on Signal Processing*, 69:1910–1923, 2021.
 - [79] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos. The variational approximation for Bayesian inference. *IEEE Signal Processing Magazine*, 25(6):131–146, 2008.
 - [80] C. Veibäck. *Tracking of animals using airborne cameras*. PhD thesis, Linköping University Electronic Press, 2016.
 - [81] G. Vivone, P. Braca, K. Granström, A. Natale, and J. Chanussot. Converted measurements Bayesian extended target tracking applied to X-band marine radar data. 2019.
 - [82] N. Wahlström and E. Özkan. Extended target tracking using Gaussian processes. *IEEE Transactions on Signal Processing*, 63(16):4165–4178, 2015.
 - [83] T. P. Wallace and P. A. Wintz. An efficient three-dimensional aircraft recognition algorithm using normalized fourier descriptors. *Computer Graphics and Image Processing*, 13(2):99–126, 1980.
 - [84] J. Xiao, R. Stolkin, and A. Leonardis. Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4978–4987, 2015.
 - [85] J. Xiao, R. Stolkin, and A. Leonardis. Dynamic multi-level appearance models and adaptive clustered decision trees for single target tracking. *Pattern Recognition*, 69:169–183, 2017.
 - [86] C. Yang, O. Tiebe, K. Shirahama, E. Lukasik, and M. Grzegorzec. Evaluating contour segment descriptors. *Machine Vision and Applications*, 28(3-4):373–391, 2017.

- [87] M. Yang, K. Kpalma, and J. Ronsin. A survey of shape feature extraction techniques. In *Pattern Recognition*, pages 43–90. IN-TECH, 2008.
- [88] S. Yang and M. Baum. Second-order extended Kalman filter for extended object and group tracking. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1178–1184. IEEE, 2016.
- [89] S. Yang and M. Baum. Extended Kalman filter for extended object tracking. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4386–4390. IEEE, 2017.
- [90] S. Yang and M. Baum. Tracking the orientation and axes lengths of an elliptical extended object. *IEEE Transactions on Signal Processing*, 67(18):4720–4729, Jul 2019.
- [91] S. Yang, M. Baum, and K. Granström. Metrics for performance evaluation of elliptic extended object tracking methods. In *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 523–528. IEEE, 2016.

APPENDICES

.1.1 Derivation of Variational Updates

In this appendix, we are going to derive the iterations which would calculate the $(i+1)$ th iterates of the distributions $q_{\mathbf{z}}(\mathbf{z}_k^{1:M_k})$, $q_{\mathbf{r}}(\mathbf{r}_k^{1:M_k})$, $q_{\mathbf{x}}(\mathbf{x}_k)$, $q_{\mathbf{X}}(\mathbf{X}_k^{1:L})$, $q_{\pi}(\pi_k^{1:L})$ from their corresponding i th iterates.

.1.1.1 Calculation of $q_{\mathbf{z}}^{(i+1)}(\cdot)$

The density $q_{\mathbf{z}}^{(i+1)}(\cdot)$ is given as [8, Ch. 10]

$$\log q_{\mathbf{z}}^{(i+1)}(\mathbf{z}_k^{1:M_k}) = \mathbb{E}_{\mathbf{z}} [\log p(\cdot | \mathcal{Y}_{k-1})] + c_{\mathbf{z}} \quad (.1)$$

where $c_{\mathbf{z}}$ denotes any constant term(s) with respect to the variables $\mathbf{z}_k^{1:M}$. The joint log-distribution in the expectation above is given as

$$\begin{aligned} \log p(\cdot | \mathcal{Y}_{k-1}) &= \sum_{j=1}^{M_k} \left(\log \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) \right. \\ &\quad \left. + \sum_{\ell=1}^L r_k^{j,\ell} \log \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_{\ell} \mathbf{x}_k, s \mathbf{X}_k^{\ell}) \right) + c_{\mathbf{z}}, \end{aligned} \quad (.2a)$$

$$\begin{aligned} &= \sum_{j=1}^{M_k} \left(\log \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) \right. \\ &\quad \left. - \frac{1}{2} \sum_{\ell=1}^L r_k^{j,\ell} (\mathbf{z}_k^j - \mathbf{H}_{\ell} \mathbf{x}_k)^T (s \mathbf{X}_k^{\ell})^{-1} (\cdot) \right) + c_{\mathbf{z}}. \end{aligned} \quad (.2b)$$

Taking the expectation of both sides, we get

$$\begin{aligned} \mathbb{E}_{\mathbf{z}} [\log p(\cdot | \mathcal{Y}_{k-1})] &= \sum_{j=1}^{M_k} \left(\log \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) \right. \\ &\quad \left. - \frac{1}{2} \sum_{\ell=1}^L \overline{r_k^{j,\ell}} (\mathbf{z}_k^j - \mathbf{H}_{\ell} \overline{\mathbf{x}}_k)^T (\overline{s \mathbf{X}_k^{\ell}})^{-1} (\cdot) \right) + c_{\mathbf{z}}, \end{aligned} \quad (.3a)$$

$$\begin{aligned}
&= \sum_{j=1}^{M_k} \left(\log \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) \right. \\
&\quad \left. - \frac{1}{2} \sum_{\ell=1}^L (\mathbf{z}_k^j - \mathbf{H}_\ell \bar{\mathbf{x}}_k)^T \overline{r_k^{j,\ell} (s\mathbf{X}_k^\ell)^{-1}} (\cdot) \right) + c_{\mathbf{z}}, \tag{.3b}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{j=1}^{M_k} \left(\log \mathcal{N}(\mathbf{y}_k^j; \mathbf{z}_k^j, \mathbf{R}) \right. \\
&\quad \left. + \sum_{\ell=1}^L \log \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_\ell \bar{\mathbf{x}}_k, \left(\overline{r_k^{j,\ell} (s\mathbf{X}_k^\ell)^{-1}} \right)^{-1}) \right) + c_{\mathbf{z}}, \tag{.3c}
\end{aligned}$$

$$= \sum_{j=1}^{M_k} \log \mathcal{N}(\mathbf{z}_k^j; \hat{\mathbf{z}}_{k|k}^{j,(i+1)}, \mathbf{P}_{k|k}^{z,j,(i+1)}) + c_{\mathbf{z}}, \tag{.3d}$$

where

$$\hat{\mathbf{z}}_{k|k}^{j,(i+1)} \triangleq \mathbf{P}_{k|k}^{z,j,(i+1)} \left(\mathbf{R}^{-1} \mathbf{y}_k^j + \sum_{\ell=1}^L \overline{r_k^{j,\ell} (s\mathbf{X}_k^\ell)^{-1}} \mathbf{H}_\ell \bar{\mathbf{x}}_k \right), \tag{.4a}$$

$$\mathbf{P}_{k|k}^{z,j,(i+1)} \triangleq \left(\mathbf{R}^{-1} + \sum_{\ell=1}^L \overline{r_k^{j,\ell} (s\mathbf{X}_k^\ell)^{-1}} \right)^{-1}, \tag{.4b}$$

for $j = 1, \dots, M_k$. Hence we have

$$q_{\mathbf{z}}^{(i+1)}(\mathbf{z}_k^{1:M_k}) = \prod_{j=1}^{M_k} \mathcal{N}(\mathbf{z}_k^j; \hat{\mathbf{z}}_{k|k}^{j,(i+1)}, \mathbf{P}_{k|k}^{z,j,(i+1)}). \tag{.5}$$

.1.1.2 Calculation of $q_{\mathbf{r}}^{(i+1)}(\cdot)$

The density $q_{\mathbf{r}}^{(i+1)}(\cdot)$ is given as

$$\log q_{\mathbf{r}}^{(i+1)}(\mathbf{r}_k^{1:M_k}) = \mathbb{E}_{\mathbf{r}} [\log p(\cdot | \mathbf{y}_{k-1})] + c_{\mathbf{r}} \tag{.6}$$

where $c_{\mathbf{r}}$ denotes any constant term(s) with respect to the variables $\mathbf{r}_k^{1:M}$. The joint log-distribution in the expectation above is given as

$$\begin{aligned}
&\log p(\cdot | \mathbf{y}_{k-1}) \\
&= \sum_{j=1}^{M_k} \sum_{\ell=1}^L r_k^{j,\ell} \left(\log \pi_k^\ell + \log \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s\mathbf{X}_k^\ell) \right) + c_{\mathbf{r}}, \tag{.7a}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{j=1}^{M_k} \sum_{\ell=1}^L r_k^{j,\ell} \left(\log \pi_k^\ell - \frac{1}{2} \log |2\pi s \mathbf{X}_k^\ell| \right. \\
&\quad \left. - \frac{1}{2} (\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k)^T (s \mathbf{X}_k^\ell)^{-1} (\cdot) \right) + c_{\setminus \mathbf{r}}, \tag{.7b}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{j=1}^{M_k} \sum_{\ell=1}^L r_k^{j,\ell} \left(\log \pi_k^\ell - \frac{1}{2} \log |2\pi s \mathbf{X}_k^\ell| \right. \\
&\quad \left. - \frac{1}{2} \text{Tr} \left((s \mathbf{X}_k^\ell)^{-1} (\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k) (\cdot)^T \right) \right) + c_{\setminus \mathbf{r}}. \tag{.7c}
\end{aligned}$$

Taking the expectation of both sides, we get

$$\begin{aligned}
&\mathbb{E}_{\setminus \mathbf{r}} [\log p(\cdot | \mathcal{Y}_{k-1})] \\
&= \sum_{j=1}^{M_k} \sum_{\ell=1}^L r_k^{j,\ell} \left(\overline{\log \pi_k^\ell} - \frac{1}{2} \overline{\log |2\pi s \mathbf{X}_k^\ell|} \right. \\
&\quad \left. - \frac{1}{2} \text{Tr} \left(\overline{(s \mathbf{X}_k^\ell)^{-1} (\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k) (\cdot)^T} \right) \right) + c_{\setminus \mathbf{r}}, \tag{.8a}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{j=1}^{M_k} \sum_{\ell=1}^L r_k^{j,\ell} \log \exp \left(\overline{\log \pi_k^\ell} - \frac{1}{2} \overline{\log |2\pi s \mathbf{X}_k^\ell|} \right. \\
&\quad \left. - \frac{1}{2} \text{Tr} \left(\overline{(s \mathbf{X}_k^\ell)^{-1} (\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k) (\cdot)^T} \right) \right) + c_{\setminus \mathbf{r}}, \tag{.8b}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{j=1}^{M_k} \sum_{\ell=1}^L \log \left(\left(\exp \left(\overline{\log \pi_k^\ell} - \frac{1}{2} \overline{\log |2\pi s \mathbf{X}_k^\ell|} \right. \right. \right. \\
&\quad \left. \left. - \frac{1}{2} \text{Tr} \left(\overline{(s \mathbf{X}_k^\ell)^{-1} (\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k) (\cdot)^T} \right) \right) \right)^{r_k^{j,\ell}} \right) + c_{\setminus \mathbf{r}}, \\
&= \sum_{j=1}^{M_k} \sum_{\ell=1}^L \log \left((\gamma_{k|k}^{j,\ell,(i+1)})^{r_k^{j,\ell}} \right) + c_{\setminus \mathbf{r}}, \tag{.8c}
\end{aligned}$$

where

$$\gamma_{k|k}^{j,\ell,(i+1)} \triangleq \frac{\tilde{\gamma}_{k|k}^{j,\ell,(i+1)}}{\sum_{\ell=1}^L \tilde{\gamma}_{k|k}^{j,\ell,(i+1)}}, \tag{.9a}$$

$$\begin{aligned}
\tilde{\gamma}_{k|k}^{j,\ell,(i+1)} &\triangleq \exp \left(\overline{\log \pi_k^\ell} - \frac{1}{2} \overline{\log |\mathbf{X}_k^\ell|} \right. \\
&\quad \left. - \frac{1}{2} \text{Tr} \left(\overline{(s \mathbf{X}_k^\ell)^{-1} (\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k) (\cdot)^T} \right) \right), \tag{.9b}
\end{aligned}$$

for $j = 1, \dots, M_k$, $\ell = 1, \dots, L$. Hence, we have

$$q_{\mathbf{r}}^{(i+1)}(\mathbf{r}_k^{1:M_k}) = \prod_{j=1}^{M_k} \prod_{\ell=1}^L (\gamma_{k|k}^{j,\ell,(i+1)})^{r_k^{j,\ell}}. \tag{.10}$$

Note that the expression above corresponds to the following probabilistic characterization

$$P_{q_{\mathbf{r}}^{(i+1)}} \{r_k^{j,\ell} = 1\} = \gamma_{k|k}^{j,\ell,(i+1)} \quad (.11)$$

for $j = 1, \dots, M_k, \ell = 1, \dots, L$.

.1.1.3 Calculation of $q_{\mathbf{x}}^{(i+1)}(\cdot)$

The density $q_{\mathbf{x}}^{(i+1)}(\cdot)$ is given as

$$\log q_{\mathbf{x}}^{(i+1)}(\mathbf{x}_k) = \mathbb{E}_{\setminus \mathbf{x}} [\log p(\cdot | \mathbf{y}_{k-1})] + c_{\setminus \mathbf{x}} \quad (.12)$$

where $c_{\setminus \mathbf{x}}$ denotes any constant terms with respect to the variable $\mathbf{x}_k$. The joint log-distribution in the expectation above is given as

$$\begin{aligned} \log p(\cdot | \mathbf{y}_{k-1}) &= \sum_{j=1}^{M_k} \sum_{\ell=1}^L r_k^{j,\ell} \log \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_{\ell} \mathbf{x}_k, s \mathbf{X}_k^{\ell}) \\ &\quad + \log \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\setminus \mathbf{x}}, \end{aligned} \quad (.13a)$$

$$\begin{aligned} &= \sum_{\ell=1}^L \sum_{j=1}^{M_k} r_k^{j,\ell} \log \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_{\ell} \mathbf{x}_k, s \mathbf{X}_k^{\ell}) \\ &\quad + \log \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\setminus \mathbf{x}}, \end{aligned} \quad (.13b)$$

$$\begin{aligned} &= -\frac{1}{2} \sum_{\ell=1}^L \sum_{j=1}^{M_k} r_k^{j,\ell} (\mathbf{z}_k^j - \mathbf{H}_{\ell} \mathbf{x}_k)^T (s \mathbf{X}_k^{\ell})^{-1} (\cdot) \\ &\quad + \log \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\setminus \mathbf{x}}, \end{aligned} \quad (.13c)$$

$$\begin{aligned} &= -\frac{1}{2} \sum_{\ell=1}^L \text{Tr} \left((s \mathbf{X}_k^{\ell})^{-1} \sum_{j=1}^{M_k} r_k^{j,\ell} (\mathbf{z}_k^j - \mathbf{H}_{\ell} \mathbf{x}_k) (\cdot)^T \right) \\ &\quad + \log \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\setminus \mathbf{x}}. \end{aligned} \quad (.13d)$$

Taking the expectation of both sides, we get

$$\mathbb{E}_{\setminus \mathbf{x}} [\log p(\cdot | \mathbf{y}_{k-1})]$$

$$\begin{aligned}
&= -\frac{1}{2} \sum_{\ell=1}^L \text{Tr} \left(\overline{(s\mathbf{X}_k^\ell)^{-1}} \sum_{j=1}^{M_k} \overline{r_k^{j,\ell}} (\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k) (\cdot)^T \right) \\
&\quad + \log \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\setminus \mathbf{x}},
\end{aligned} \tag{.14a}$$

$$\begin{aligned}
&= -\frac{1}{2} \sum_{\ell=1}^L \text{Tr} \left(\overline{(s\mathbf{X}_k^\ell)^{-1}} \left(\sum_{j=1}^{M_k} \overline{r_k^{j,\ell}} \right) \right. \\
&\quad \times \left. \frac{\sum_{j=1}^{M_k} \overline{r_k^{j,\ell}} (\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k) (\cdot)^T}{\sum_{j=1}^{M_k} \overline{r_k^{j,\ell}}} \right) \\
&\quad + \log \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\setminus \mathbf{x}},
\end{aligned} \tag{.14b}$$

$$\begin{aligned}
&= -\frac{1}{2} \sum_{\ell=1}^L \text{Tr} \left(\overline{(s\mathbf{X}_k^\ell)^{-1}} \left(\sum_{j=1}^{M_k} \overline{r_k^{j,\ell}} \right) \right. \\
&\quad \times \left(\frac{\sum_{j=1}^{M_k} \overline{r_k^{j,\ell}} \mathbf{z}_k^j}{\sum_{j=1}^{M_k} \overline{r_k^{j,\ell}}} - \mathbf{H}_\ell \mathbf{x}_k \right) \left(\cdot \right)^T \Bigg) \\
&\quad + \log \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\setminus \mathbf{x}},
\end{aligned} \tag{.14c}$$

$$\begin{aligned}
&= -\frac{1}{2} \sum_{\ell=1}^L \text{Tr} \left(\overline{(s\mathbf{X}_k^\ell)^{-1}} \left(\sum_{j=1}^{M_k} \overline{r_k^{j,\ell}} \right) (\mathbf{u}_k^\ell - \mathbf{H}_\ell \mathbf{x}_k) (\cdot)^T \right) \\
&\quad + \log \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\setminus \mathbf{x}}, \\
&= \sum_{\ell=1}^L \log \mathcal{N}(\mathbf{u}_k^\ell; \mathbf{H}_\ell \mathbf{x}_k, (\mathbf{\Lambda}_k^\ell)^{-1}) \\
&\quad + \log \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\setminus \mathbf{x}},
\end{aligned} \tag{.14d}$$

$$\begin{aligned}
&= \log \mathcal{N}(\mathbf{u}_K^{1:L}; \mathbf{H}_{1:L} \mathbf{x}_k, \mathbf{\Lambda}_{1:L}^{-1}) \\
&\quad + \log \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) + c_{\setminus \mathbf{x}},
\end{aligned} \tag{.14e}$$

where

$$\mathbf{u}_k^\ell \triangleq \frac{\sum_{j=1}^{M_k} \overline{r_k^{j,\ell}} \mathbf{z}_k^j}{\sum_{j=1}^{M_k} \overline{r_k^{j,\ell}}}, \tag{.15a}$$

$$\mathbf{\Lambda}_k^\ell \triangleq \overline{(s\mathbf{X}_k^\ell)^{-1}} \sum_{j=1}^{M_k} \overline{r_k^{j,\ell}}, \tag{.15b}$$

$$\mathbf{u}_k^{1:L} \triangleq [(\mathbf{u}_k^1)^T \quad (\mathbf{u}_k^2)^T \quad \cdots \quad (\mathbf{u}_k^L)^T]^T, \tag{.15c}$$

$$\mathbf{H}_{1:L} \triangleq [\mathbf{H}_1^T \quad \mathbf{H}_2^T \quad \cdots \quad \mathbf{H}_L^T]^T, \tag{.15d}$$

$$\mathbf{\Lambda}_{1:L} \triangleq \text{blkdiag}(\mathbf{\Lambda}_k^1, \mathbf{\Lambda}_k^2, \dots, \mathbf{\Lambda}_k^L). \quad (.15e)$$

for $\ell = 1, \dots, L$. Hence we have

$$q_{\mathbf{x}}^{(i+1)}(\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k}^{(i+1)}, \mathbf{P}_{k|k}^{(i+1)}), \quad (.16)$$

where

$$\mathbf{m}_{k|k}^{(i+1)} = \mathbf{P}_{k|k}^{(i+1)} (\mathbf{P}_{k|k-1}^{-1} \mathbf{m}_{k|k-1} + \mathbf{H}_{1:L}^T \mathbf{\Lambda}_{1:L} \mathbf{u}_k^{1:L}), \quad (.17a)$$

$$\mathbf{P}_{k|k}^{(i+1)} = (\mathbf{P}_{k|k-1}^{-1} + \mathbf{H}_{1:L}^T \mathbf{\Lambda}_{1:L} \mathbf{H}_{1:L})^{-1}. \quad (.17b)$$

.1.1.4 Calculation of $q_{\mathbf{x}}^{(i+1)}(\cdot)$

The density $q_{\mathbf{x}}^{(i+1)}(\cdot)$ is given as

$$\log q_X^{(i+1)}(\mathbf{X}_k^{1:L}) = \mathbb{E}_{\setminus \mathbf{x}} [\log p(\cdot | \mathbf{y}_{k-1})] + c_{\setminus \mathbf{x}} \quad (.18)$$

where $c_{\setminus \mathbf{x}}$ denotes any constant terms with respect to the variables $\mathbf{X}_k^{1:L}$. The joint log-distribution in the expectation above is given as

$$\begin{aligned} \log p(\cdot | \mathbf{y}_{k-1}) &= \sum_{j=1}^{M_k} \sum_{\ell=1}^L r_k^{j,\ell} \log \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s \mathbf{X}_k^\ell) \\ &\quad + \sum_{\ell=1}^L \log \mathcal{IW}(\mathbf{X}_k^\ell; v_{k|k-1}^\ell, \mathbf{V}_{k|k-1}^\ell) + c_{\setminus \mathbf{x}}, \end{aligned} \quad (.19a)$$

$$\begin{aligned} &= \sum_{\ell=1}^L \left(\sum_{j=1}^{M_k} r_k^{j,\ell} \log \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_\ell \mathbf{x}_k, s \mathbf{X}_k^\ell) \right. \\ &\quad \left. + \log \mathcal{IW}(\mathbf{X}_k^\ell; v_{k|k-1}^\ell, \mathbf{V}_{k|k-1}^\ell) \right) + c_{\setminus \mathbf{x}}, \end{aligned} \quad (.19b)$$

$$\begin{aligned} &= \sum_{\ell=1}^L \left(-\frac{1}{2} \left(\sum_{j=1}^{M_k} r_k^{j,\ell} \right) \log |\mathbf{X}_k^\ell| \right. \\ &\quad - \frac{1}{2} \text{Tr} \left((\mathbf{X}_k^\ell)^{-1} \frac{1}{s} \sum_{j=1}^M r_k^{j,\ell} (\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k) (\cdot)^T \right) \\ &\quad \left. + \log \mathcal{IW}(\mathbf{X}_k^\ell; v_{k|k-1}^\ell, \mathbf{V}_{k|k-1}^\ell) \right) + c_{\setminus \mathbf{x}}. \end{aligned} \quad (.19c)$$

Taking the expectation of both sides, we get

$$\begin{aligned}
& \mathbb{E}_{\mathbf{X}} [\log p(\cdot | \mathcal{Y}_{k-1})] \\
&= \sum_{\ell=1}^L \left(-\frac{1}{2} \left(\sum_{j=1}^{M_k} \overline{r_k^{j,\ell}} \right) \log |\mathbf{X}_k^\ell| \right. \\
&\quad \left. - \frac{1}{2} \text{Tr} \left((\mathbf{X}_k^\ell)^{-1} \frac{1}{s} \sum_{j=1}^M \overline{r_k^{j,\ell}} (\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k) (\cdot)^T \right) \right. \\
&\quad \left. + \log \mathcal{IW}(\mathbf{X}_k^\ell; v_{k|k-1}^\ell, \mathbf{V}_{k|k-1}^\ell) \right) + c_{\mathbf{X}}, \tag{.20a}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{\ell=1}^L \log \mathcal{IW} \left(\mathbf{X}_k^\ell; \sum_{j=1}^{M_k} \overline{r_k^{j,\ell}}, \frac{1}{s} \sum_{j=1}^M \overline{r_k^{j,\ell}} (\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k) (\cdot)^T \right) \\
&\quad + \log \mathcal{IW}(\mathbf{X}_k^\ell; v_{k|k-1}^\ell, \mathbf{V}_{k|k-1}^\ell) + c_{\mathbf{X}}, \\
&= \sum_{\ell=1}^L \log \mathcal{IW}(\mathbf{X}_k^\ell; v_{k|k}^{\ell,(i+1)}, \mathbf{V}_{k|k}^{\ell,(i+1)}) + c_{\mathbf{X}}, \tag{.20b}
\end{aligned}$$

where

$$v_{k|k}^{\ell,(i+1)} \triangleq v_{k|k-1}^\ell + \sum_{j=1}^{M_k} \overline{r_k^{j,\ell}}, \tag{.21a}$$

$$\mathbf{V}_{k|k}^{\ell,(i+1)} \triangleq \mathbf{V}_{k|k-1}^\ell + \frac{1}{s} \sum_{j=1}^M \overline{r_k^{j,\ell}} (\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k) (\cdot)^T, \tag{.21b}$$

for $\ell = 1, \dots, L$. Hence, we have

$$q_X^{(i+1)}(\mathbf{X}_k^{1:L}) = \prod_{\ell=1}^L \mathcal{IW}(\mathbf{X}_k^\ell; v_{k|k}^{\ell,(i+1)}, \mathbf{V}_{k|k}^{\ell,(i+1)}). \tag{.22}$$

.1.1.5 Calculation of $q_\pi^{(i+1)}(\cdot)$

The density $q_\pi^{(i+1)}(\cdot)$ is given as

$$\log q_\pi^{(i+1)}(\pi_k^{1:L}) = \mathbb{E}_{\mathbf{X}} [\log p(\cdot | \mathcal{Y}_{k-1})] + c_{\mathbf{X}} \tag{.23}$$

where $c_{\mathbf{X}}$ denotes any constant terms with respect to the variables $\pi_k^{1:L}$. The joint log-distribution in the expectation above is given as

$$\log p(\cdot | \mathcal{Y}_{k-1})$$

$$\begin{aligned}
&= \left(\sum_{j=1}^{M_k} \sum_{\ell=1}^L r_k^{j,\ell} \log \pi_k^\ell \right) \\
&+ \log D(\pi_k^{1:L}; \{\alpha_{k|k-1}^\ell\}_{\ell=1}^L) + c_{\setminus \pi},
\end{aligned} \tag{.24a}$$

$$= \sum_{j=1}^{M_k} \sum_{\ell=1}^L r_k^{j,\ell} \log \pi_k^\ell + \sum_{\ell=1}^L (\alpha_{k|k-1}^\ell - 1) \log \pi_k^\ell + c_{\setminus \pi}, \tag{.24b}$$

$$= \sum_{\ell=1}^L \left(\sum_{j=1}^{M_k} r_k^{j,\ell} + \alpha_{k|k-1}^\ell - 1 \right) \log \pi_k^\ell + c_{\setminus \pi}. \tag{.24c}$$

Taking the expectation of both sides, we get

$$\begin{aligned}
&\mathbb{E}_{\setminus \pi} [\log p(\cdot | \mathbf{y}_{k-1})] \\
&= \sum_{\ell=1}^L \left(\sum_{j=1}^{M_k} \overline{r_k^{j,\ell}} + \alpha_{k|k-1}^\ell - 1 \right) \log \pi_k^\ell + c_{\setminus \pi}.
\end{aligned} \tag{.25}$$

Hence we have

$$q_\pi^{(i+1)}(\pi_k^{1:L}) = D(\pi_k^{1:L}; \{\alpha_{k|k}^{\ell,(i+1)}\}_{\ell=1}^L) \tag{.26}$$

where

$$\alpha_{k|k}^{\ell,(i+1)} \triangleq \alpha_{k|k-1}^\ell + \sum_{j=1}^{M_k} \overline{r_k^{j,\ell}}, \tag{.27}$$

for $\ell = 1, \dots, L$.

.1.2 Expected Values Required for Iterations

The expected values required in the previous subsections can be calculated as

$$\begin{aligned}
\overline{(s\mathbf{X}_k^\ell)^{-1}} &\triangleq \mathbb{E}_{\mathbf{X}} [(s\mathbf{X}_k^\ell)^{-1}] \\
&= (v_{k|k}^{\ell,(i)} - n_y - 1) (s\mathbf{V}_{k|k}^{\ell,(i)})^{-1},
\end{aligned} \tag{.28a}$$

$$\overline{\mathbf{x}_k} \triangleq \mathbb{E}_{\mathbf{X}} [\mathbf{x}_k] = \mathbf{m}_{k|k}^{(i)}, \tag{.28b}$$

$$\overline{r_k^{j,\ell}} \triangleq \mathbb{E}_{\mathbf{r}} [r_k^{j,\ell}] = P_{q_{\mathbf{r}}^{(i)}} \{r_k^{j,\ell} = 1\} = \gamma_{k|k}^{j,\ell,(i)}, \tag{.28c}$$

$$\begin{aligned}
\overline{\log \pi_k^\ell} &\triangleq E_\pi [\log \pi_k^\ell] \\
&= \psi(\alpha_{k|k}^{\ell,(i)}) - \psi\left(\sum_{\ell=1}^L \alpha_{k|k}^{\ell,(i)}\right),
\end{aligned} \tag{.28d}$$

$$\begin{aligned}
\overline{\log |\mathbf{X}_k^\ell|} &\triangleq \mathbb{E}_{\mathbf{X}} [\log |\mathbf{X}_k^\ell|] \\
&= \log |\mathbf{V}_{k|k}^{\ell,(i)}| - n_y \log 2 \\
&\quad - \sum_{d=1}^{n_y} \psi \left(\frac{v_{k|k}^{\ell,(i)} - n_y - d}{2} \right),
\end{aligned} \tag{.28e}$$

$$\overline{\mathbf{z}_k^j} \triangleq \mathbb{E}_{\mathbf{z}} [\mathbf{z}_k^j] = \mathbf{z}_{k|k}^{j,(i)}, \tag{.28f}$$

$$\begin{aligned}
\overline{(\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k)(\cdot)^T} &\triangleq \mathbb{E}_{\mathbf{z}, \mathbf{x}} [(\mathbf{z}_k^j - \mathbf{H}_\ell \mathbf{x}_k)(\cdot)^T] \\
&= (\mathbf{z}_{k|k}^{j,(i)} - \mathbf{H}_\ell \mathbf{m}_{k|k}^{(i)})(\cdot)^T + \mathbf{P}_{k|k}^{z,j,(i)} \\
&\quad + \mathbf{H}_\ell \mathbf{P}_{k|k}^{(i)} \mathbf{H}_\ell^T,
\end{aligned} \tag{.28g}$$

for $j = 1, \dots, M_k$, $\ell = 1, \dots, L$ where $\psi(\cdot)$ denotes the digamma function (i.e., the logarithmic derivative of the Gamma function or polygamma function of order zero.).

.1.3 Calculation of δ and Δ

In Section 3.3.5, we introduced the update formulas for the orientation distribution as below.

$$\hat{\theta}_{k|k}^{(\ell+1)} = \Theta_{k|k}^{(\ell+1)} (\Theta_{k|k-1}^{-1} \hat{\theta}_{k|k-1} + \delta), \tag{.29}$$

$$\Theta_{k|k}^{(\ell+1)} = (\Theta_{k|k-1}^{-1} + \Delta)^{-1}, \tag{.30}$$

Here, the variables δ and Δ are derived from the expectation in (.31).

$$\frac{-1}{2} \sum_{j=1}^{m_k} \mathbb{E}_{\theta_k} [(a - b\theta_k)^T (sX)^{-1} (a - b\theta_k)], \tag{.31}$$

where δ and Δ are denoted as

$$\delta \triangleq \sum_{j=1}^{m_k} \text{Tr} \left[\mathbb{E}_{q_X^{(\ell)}} [(sX_k)^{-1}] \mathbb{E}_{q_{\mathbf{x}}^{(\ell)}, q_{\mathbf{z}}^{(\ell)}} [ab^T] \right], \tag{.32}$$

$$\Delta \triangleq \sum_{j=1}^{m_k} \text{Tr} \left[\mathbb{E}_{q_X^{(\ell)}} [(sX_k)^{-1}] \mathbb{E}_{q_{\mathbf{x}}^{(\ell)}, q_{\mathbf{z}}^{(\ell)}} [bb^T] \right], \tag{.33}$$

and

$$a = (T_{\hat{\theta}_{k|k}^{(\ell)}})^T (\mathbf{z}_k^j - H \mathbf{x}_k) - (T'_{\hat{\theta}_{k|k}^{(\ell)}})^T (\mathbf{z}_k^j - H \mathbf{x}_k) \hat{\theta}_{k|k}^{(\ell)}, \tag{.34}$$

$$b = -(T'_{\hat{\theta}_{k|k}^{(\ell)}})^T (\mathbf{z}_k^j - H\mathbf{x}_k). \quad (.35)$$

When the a and b variables are substituted into (.32) and (.33), we obtain the δ and Δ variables as

$$\begin{aligned} \delta = \sum_{j=1}^{m_k} \text{Tr} \left[\overline{sX_k^{-1} (T'_{\hat{\theta}_{k|k}^{(\ell)}})^T (\mathbf{z}_k^j - H\mathbf{x}_k) (\cdot)^T} (T'_{\hat{\theta}_{k|k}^{(\ell)}}) \hat{\theta}_{k|k}^{(\ell)} \right] \\ - \text{Tr} \left[\overline{sX_k^{-1} T_{\hat{\theta}_{k|k}^{(\ell)}}^T (\mathbf{z}_k^j - H\mathbf{x}_k) (\cdot)^T} (T'_{\hat{\theta}_{k|k}^{(\ell)}}) \right], \end{aligned} \quad (.36)$$

$$\Delta = \sum_{j=1}^{m_k} \text{Tr} \left[\overline{sX_k^{-1} (T'_{\hat{\theta}_{k|k}^{(\ell)}})^T (\mathbf{z}_k^j - H\mathbf{x}_k) (\cdot)^T} (T'_{\hat{\theta}_{k|k}^{(\ell)}}) \right], \quad (.37)$$

where $\overline{(\mathbf{z}_k^j - H\mathbf{x}_k) (\cdot)^T} = \mathbb{E}_{q_{\mathbf{x}}^{(\ell)}, q_{\mathbf{z}}^{(\ell)}} \left[(\mathbf{z}_k^j - H\mathbf{x}_k) (\cdot)^T \right]$, and $\overline{sX_k^{-1}} = \mathbb{E}_{q_X^{(\ell)}} [(sX_k)^{-1}]$.

.1.4 Proof of Lemma 1

In this section we will give the proof of *Lemma 1* to calculate $\mathbb{E}_{q_{\theta}^{(\ell)}} [(sT_{\theta_k} M T_{\theta_k}^T)^{-1}]$. In the formulation, we first multiply the matrices inside the expectation. Then, the expectation of each entry of the resultant matrix is taken. First, notice that,

$$(T_{\theta_k} M T_{\theta_k}^T)^{-1} = T_{\theta_k} M^{-1} T_{\theta_k}^T. \quad (.38)$$

Given

$$M^{-1} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix},$$

the expression whose expectation has to be taken becomes

$$\begin{aligned} \Lambda &= T_{\theta_k} M^{-1} T_{\theta_k}^T \\ &= \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \\ &\quad \times \begin{bmatrix} \cos(\theta_k) & \sin(\theta_k) \\ -\sin(\theta_k) & \cos(\theta_k) \end{bmatrix}, \end{aligned} \quad (.39a)$$

$$\Lambda(1, 1) = m_{11} \cos^2(\theta_k) + m_{22} \sin^2(\theta_k)$$

$$- (m_{12} + m_{21}) \cos(\theta_k) \sin(\theta_k), \quad (.39b)$$

$$\begin{aligned} \Lambda(1, 2) &= m_{12} \cos^2(\theta_k) - m_{21} \sin^2(\theta_k) \\ &\quad + (m_{11} - m_{22}) \cos(\theta_k) \sin(\theta_k), \end{aligned} \quad (.39c)$$

$$\begin{aligned} \Lambda(2, 1) &= m_{21} \cos^2(\theta_k) - m_{12} \sin^2(\theta_k) \\ &\quad + (m_{11} - m_{22}) \cos(\theta_k) \sin(\theta_k), \end{aligned} \quad (.39d)$$

$$\begin{aligned} \Lambda(2, 2) &= m_{22} \cos^2(\theta_k) + m_{11} \sin^2(\theta_k) \\ &\quad + (m_{12} + m_{21}) \cos(\theta_k) \sin(\theta_k). \end{aligned} \quad (.39e)$$

Now the following trigonometric transformations are utilized and the expectations are taken with respect to the resulting expressions.

$$\cos^2(\theta_k) = \frac{1 + \cos(2\theta_k)}{2}, \quad \sin^2(\theta_k) = \frac{1 - \cos(2\theta_k)}{2}, \quad (.40a)$$

$$\cos(\theta_k) \sin(\theta_k) = \frac{\sin(2\theta_k)}{2}, \quad (.40b)$$

$$\mathbb{E}_{q_\theta^{(\ell)}} [\cos(2\theta_k)] = \cos(2\hat{\theta}_{k|k}^{(\ell)} \exp(-2\Theta_{k|k}^{(\ell)})), \quad (.40c)$$

$$\mathbb{E}_{q_\theta^{(\ell)}} [\sin(2\theta_k)] = \sin(2\hat{\theta}_{k|k}^{(\ell)} \exp(-2\Theta_{k|k}^{(\ell)})). \quad (.40d)$$

By substituting the expressions in (.39) with the corresponding equalities given in (.40) *Lemma 1* is obtained.

.1.5 The parameters of the experiment in section 3.5

In this section, the parameters of the experiment given in Section 3.5 are summarized. The prior shape estimate is apart from the true center location of the target by (20, 20) units in the 2D coordinate frame. The extent of the ground truth object is parametrized as $\text{diag}([6, 0.5])$ with 0° orientation. The mean of the prior extent ellipse parameters for both methods are equal to the parameters of the ground truth extent. The prior shape parameters of the proposed algorithm are selected to be $\alpha_0^{1,2} = [101 \ 101]^T$ and $\beta_0^{1,2} = [600 \ 50]^T$. The prior mean of the orientation variable is taken as $\hat{\theta}_0 = \frac{\pi}{2}$. To have a reasonable comparison, the mean vector and the variance of the shape parameters for the EKF approach are selected to match with those of the VB algorithm. The prior kinematic state covariance matrix is $\mathbf{P}_0 = \text{diag}([300, 300, 1, 1, \frac{\pi}{2}])$. The mea-

surement noise covariance matrix is $R = \text{diag}([1, 1])$. In this experiment, the number of measurements is 10, and the measurements are generated according to a Gaussian distribution. However, the trials with the uniformly distributed measurements yield similar results.

.1.6 An example for the collapsing extent estimates

Here we repeated the simulation in Section 3.5 with the following set of parameters:

For VB, the prior mean of the target's kinematic state is $\hat{\mathbf{x}}_0 = [-30 \ -30 \ 1 \ 1 \ \frac{\pi}{2}]^T$. The prior kinematic state covariance matrix is $P_0 = \text{diag}([300, 300, 1, 1, \frac{\pi}{2}])$. The shape parameters for VB is $\alpha^{1,2} = [11 \ 11]^T$ and $\beta^{1,2} = [600 \ 50]^T$. For Algorithm-1, the prior mean vector and covariance matrix of the kinematic state is $\hat{\mathbf{r}}_0 = \hat{\mathbf{x}}_0(1 : 4)$ and $C_0^r = \text{diag}([300, 300, 1, 1])$, respectively. The prior mean and variance of the extent parameters are the same for both algorithms. The number of measurements generated from the target is 10. The measurement noise covariance matrix is $\mathbf{R} = \text{diag}([1, 1])$. A single measurement update of VB and Algorithm-1 is visualized in Figure .1.

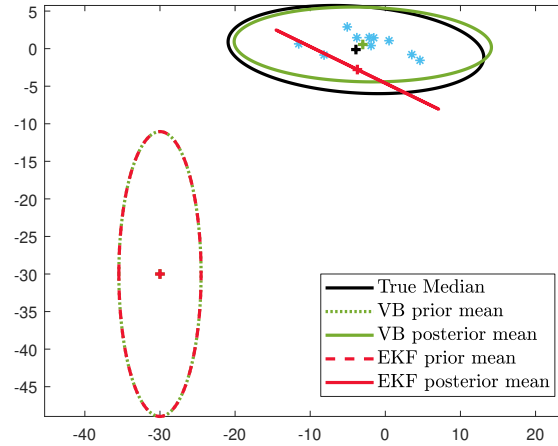


Figure .1: The visualization of the collapsing behavior of Algorithm-1. Blue stars represent the measurements, the solid green and red lines stand for the posterior means of the VB and EKF updates, respectively. The solid black line indicates the median of the true posterior, which is computed by using 1 million Monte Carlo samples.