Collaborative simultaneous localization and mapping for autonomous vehicles

Diluka Moratuwage

2013

Diluka Moratuwage. (2013). Collaborative simultaneous localization and mapping for autonomous vehicles. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/55241>

<https://doi.org/10.32657/10356/55241>

**COLLABORATIVE SIMULTANEOUS
LOCALIZATION AND MAPPING FOR
AUTONOMOUS VEHICLES**

DILUKA MORATUWAGE

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2013

Acknowledgment

First and foremost I would like to acknowledge the help and support I received from late Prof. Wijesoma, without whom this wouldn't have been possible. Next I would like to convey my sincere gratitude to my current supervisor Prof. Wang Danwei, who agreed to accept me as one of his students and provided numerous technical and non-technical support since 2011.

I'm also grateful to Prof. Ba-Ngu Vo for providing numerous technical and non-technical help and support throughout my PhD candidature. His continuous support provided me the confidence to delve deep into my research problem.

I would also like to express my gratitude to Dr. Lee Kwang Wee, whom I considered as a mentor throughout my PhD studentship.

I would like to thank my dearest parents for their love and support given to me through the years, nothing would have been possible without them.

I cannot forget the help and support I received from Mr. Chia Chiu Kiat, our lab technician who provided tremendous support in many ways. Without him this work would have been much more difficult.

Finally, I must acknowledge the numerous help and support I received from my dearest friends Namal, Akshay, Nuwan, Wajira, Suresh Kumar, Manoj and Manjula. Without them my stay in Singapore would have been difficult.

Contents

Acknowledgements	iii
List of contents	v
Summary	ix
List of figures	xii
List of Acronyms	xx
Nomenclature	xxi
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Objective and Approach	2
1.3 Contributions	3
1.4 Organization of the thesis	4
2 Literature Review	6

2.1	The SLAM problem	6
2.2	The mono-SLAM problem	7
2.2.1	Problem formulation	7
2.2.2	State of the art mono-SLAM solutions	9
2.3	Collaborative Multi-Vehicle SLAM algorithms	29
2.3.1	Problem formulation	30
2.3.2	State of the art CMSLAM solutions	31
2.4	Conclusion	34
3	Collaborative Multi-Vehicle SLAM using Random Finite Sets	36
3.1	Introduction	36
3.2	Formulation of Probabilistic Random Finite Set Multi-Vehicle SLAM	37
3.3	RFS formulation of measurement and landmark map models	41
3.4	Landmark Map Posterior Estimation	45
3.4.1	PHD filter for map posterior estimation	46
3.5	Implementation of the Multi-Vehicle SLAM algorithm	51
3.5.1	Particle filter Implementation	52
3.5.2	GM-PHD filter implementation	55
3.6	Multi-Vehicle SLAM with an arbitrary number of vehicles	60
3.6.1	PHD filter for map posterior estimation	61
3.7	Computational Complexity	64

3.8	Results and Analysis	66
3.8.1	Simulation Results	66
3.8.2	Experimental Results	75
3.9	Conclusion	78
4	Collaborative Multi-Vehicle SLAM with Moving Object Tracking	80
4.1	Introduction	80
4.2	Formulation of Probabilistic Random Finite Set CMSLAM with Moving Object Tracking	83
4.3	RFS formulation of measurement and feature map models	89
4.4	Estimation of Static and Dynamic Feature Map Posteriors	92
4.4.1	PHD filter for Augmented Map posterior estimation	94
4.4.2	Extraction of static and dynamic features	97
4.5	Vehicle Trajectories Posterior Estimation	99
4.6	Results and Analysis	102
4.7	Conclusion	114
5	A Hierarchical Solution to the Collaborative Multi-Vehicle SLAM Problem	116
5.1	Introduction	116
5.2	Formulation of Probabilistic Random Finite Set HCMSLAM solution	118
5.3	Hierarchical Multi-Vehicle SLAM System Architecture	121

5.4	RFS formulation of Measurement and Landmark Map Models	126
5.5	Global Landmark Map Posterior Estimation	131
5.5.1	PHD filter for Global map posterior estimation	132
5.6	Implementation of the HCMSLAM algorithm	135
5.6.1	Particle filter Implementation	136
5.6.2	GM-PHD filter implementation	139
5.7	Simulation Results	142
5.8	Conclusion	147
6	Conclusion and Future Work	149
6.1	Summary	149
6.2	Conclusion	152
6.3	Future research directions	153
A	GM-PHD filter in RFS CMSLAM	158
	Bibliography	165

Summary

Among today's autonomous robotic applications, exploration missions in dynamic, unstructured and uncertain environmental conditions are quite common. Autonomous multi-vehicle systems come in handy for such exploration missions involving larger environments due to higher efficiency, reliability and robustness that can be achieved compared to a single autonomous vehicle.

In the absence of a-priori feature maps, various simultaneous localization and mapping (SLAM) algorithms are used in order to improve the navigation accuracy of single-vehicle systems as a key prerequisite for exploration and mapping missions. As for multi-vehicle systems, this process is called collaborative multi-vehicle simultaneous localization and mapping (CMSLAM). In multi-vehicle systems, additional scaling problems such as consistent data fusion, bandwidth constraints and computational requirements have to be taken into account in order to be used in practical applications.

The most common single-vehicle SLAM (mono-SLAM) formulation is stochastic SLAM, in which the vehicle state and landmark locations are stored in a single vector, which is propagated as a joint posterior via a recursive prediction and update (observation) process, where sensor and vehicle dynamics are modeled in state space form. In this approach, each recursion involves solving certain additional, but crucial sub-problems outside the Bayes recursion, such as clutter filtering, data

association and map management. Moreover, these conventional solutions assume several key simplifications which limit their practical applicability. The most notable assumptions are clutter free measurements, ideal landmark detection and data association, and static landmarks.

Several CMSLAM algorithms have been developed in the literature by extending such mono-SLAM solutions. Unfortunately almost all those algorithms inherit above mentioned issues from their single-vehicle counterpart. In the recent years an alternative solution based on the finite set statistics (FISST) has been proposed as a radically different alternative solution to the mono-SLAM problem. In this approach, the landmark map and the measurements are represented as random finite sets (RFS), and a joint posterior containing the landmark map and the vehicle state is propagated while catering for landmark detection uncertainty, landmark survival uncertainty, data association, map management and measurement clutter filtering within the SLAM filter framework. In this thesis we propose to extend this framework in order to solve the more general CMSLAM problem.

The first contribution of this thesis is two new CMSLAM algorithms, which are developed using the RFS SLAM framework. The proposed formulation is based on the factorization of the full CMSLAM posterior into a product of the vehicle trajectories posterior and the landmark map posterior conditioned on the vehicle trajectories. The vehicle trajectories are estimated using a Rao-Blackwellized particle filter. The landmark map and the measurements are represented as RFSs; their dynamics are appropriately modeled using FISST; and the landmark map posterior conditioned on the vehicle trajectories is estimated using a Gaussian Mixture (GM) implementation of a probability hypothesis density (PHD) filter. The performance of the proposed solutions is evaluated and presented by benchmarking against a FastSLAM based CMSLAM solution using a simulation and a practical experiment.

Secondly, a novel algorithm is presented to address the collaborative multi-vehicle SLAM with moving object tracking (CMSLAMMOT) problem by extending the proposed RFS CMSLAM solution. Although numerous SLAM algorithms exist in the robotics literature, all such algorithms assume that the environment is static and the moving objects present in the environment are considered as clutter. As a result, a separate higher level algorithm is used in order to detect and track the moving objects present in the environment for safe navigation. The proposed CMSLAMMOT solution addresses this issue and presents a novel solution which detects and tracks moving objects present in the sensor field of views (FOVs) while simultaneously performing CMSLAM. Performance of the proposed solution is evaluated using a simulation and a practical experiment.

As the third contribution, a novel hierarchical collaborative multi-vehicle SLAM algorithm (HCMSLAM) is presented by introducing a RFS mono-SLAM based sub-mapping process in order to reduce the communication bandwidth and heavy computational load at the fusion node. Instead of communicating the control and measurement data from each vehicle at each time step, the resultant sub-map along with the sequence of applied control commands during the sub-mapping process are communicated into a higher level HCMSLAM algorithm in order to jointly propagate the landmark map and the vehicle trajectories. The landmark map and the measurements are modeled as RFSs and the full CMSLAM posterior is evaluated by factorizing into a product of the vehicle trajectories posterior and the landmark map posterior conditioned on the vehicle trajectories. The vehicle trajectories are propagated using a Rao-Blackwellized particle filter and the landmark map conditioned on the vehicle trajectories is propagated using a GM implementation of a PHD filter. Simulation results are used in order to evaluate the performance of the proposed solution.

List of Figures

- 2.1 How PHD of a landmark map evolves over time in a single vehicle RFS SLAM scenario. The figures on the left shows the map of the environment with the vehicle, where black stars are the actual landmarks and the black circles denote their estimations. The figures on the right shows the corresponding PHD at that time step. 28
- 2.2 In a RFS SLAM scenario, the landmark maps correspond to four particles are shown here. Actual vehicle trajectory of each particle is shown in red and the estimated trajectory is shown in blue. Actual landmark locations are shown as black stars, and their estimates are shown as black circles. The particle with the highest weight (particle 2) is chosen as the best estimate of the vehicle trajectory and the corresponding map is chosen as the best estimate of the landmark map. 29
- 3.1 Proposed CMSLAM solution. A particle filter is used to estimate the joint vehicle trajectories. A GM-PHD filter is used to fuse the observations collected by individual collaborating robot and produce a single global map. The resultant map is then used to estimate the joint vehicle trajectories of the collaborating vehicles. 41

3.2	How PHD of the landmark map evolves over time, to recover the landmark map. In maps, black stars denote actual landmarks and the black circles denote their estimations. The yellow circles in the PHDs denote the actual landmark locations, and dotted lines denote the observations from each vehicle.	59
3.3	A mutual feature observation scenario. Blue triangles denote two robots while yellow stars denote features present in the environment. Both robots observe a common feature and produce corresponding observations $z_{k,2}^{(1)}$ and $z_{k,1}^{(2)}$	65
3.4	Ground truth of the vehicle trajectories (vehicle 1 in blue and vehicle 2 in black) with randomly placed landmarks (black stars) superimposed on the set of accumulated false measurements (in green) recorded during a sample run under the clutter rate $\lambda_c^{(r)} = 5$, where $r = 1, 2$ denotes the vehicle number. Each vehicle starts at the origin, and closes the loop. The number of landmarks deployed in the simulation environment is 81.	67
3.5	Estimated vehicle trajectories (vehicle 1 in green and vehicle 2 in red) along with estimated feature locations (in cyan circles). A considerable amount of multiple feature detections (specially when the vehicle trajectories are overlapped) can be seen in the map. Moreover, estimated vehicle trajectories have large deviations from the ground truth (in blue and black).	69

3.6	Estimated vehicle trajectories (vehicle 1 in green and vehicle 2 in red) along with estimated feature locations (in cyan circles). Few multiple feature detections can be seen in the joint sensor FOV. Estimated vehicle trajectories have small deviations from the ground truth (in blue and black).	69
3.7	Estimated vehicle trajectories (vehicle 1 in green and vehicle 2 in red) along with estimated feature locations (in cyan circles). Much more consistent estimates of the landmarks can be seen everywhere in the map. It is also apparent that the estimated vehicle trajectories tends to produce much better results in the overlapped vehicle trajectories.	70
3.8	RMS Vehicle position errors of 50 MC runs. Proposed RFS-CMSLAM solutions produce much smaller loop closing errors while the benchmarked RB-CMSLAM solution produces much larger errors under the same clutter conditions.	71
3.9	Running time of each algorithm at each simulation time step.	71
3.10	Average number of landmarks in the map, superimposed on the actual number of landmarks (in black). It can be seen that the benchmarked RB-CMSLAM algorithm produces a comparatively larger number of landmarks, while the proposed RFS-CMSLAM solutions produces much closer estimates to the number of actually observed landmarks.	72
3.11	Average map error in terms of OSPA distance. Both RFS-CMSLAM solutions produce smaller map errors compared to the benchmarked RB-CMSLAM solution. It can also be seen that RFS-CMSLAM-II solution outperforms RFS-CMSLAM-I solution due to the much consistent feature estimation in the overlapped vehicle trajectories. .	73

- 3.12 A comparison of RMS Vehicle position errors with RFS Single Vehicle SLAM (RFS-MONO-SLAM) of 50 MC runs. It can be seen that both proposed CMSLAM solutions produce slightly small RMS vehicle position errors compared to the single vehicle SLAM solution, which is mainly due to the overlapping areas of the simulation environment. 74
- 3.13 A picture of one of the robots used for the experiment. Each robot is equipped with a single axis gyroscope (KVH DSP 3000), a Laser scanner (SICK LMS 291) and wheel encoders. The control and measurement data are communicated into a centralized computer using a wireless network connection (Bullet2HP). 75
- 3.14 A picture of the robots taken during the experiment. The pillars present in the picture are used as features. Both vehicles are simultaneously driven on two vehicle trajectories and the control and observation data from each vehicle are communicated into a centralized computer. 76
- 3.15 A comparison of estimated vehicle trajectories (vehicle 1 in green and vehicle 2 in red), superimposed on a set of known way-points (vehicle 1 in green crosses and vehicle 2 in red crosses) on which the vehicles were driven. Estimated feature locations (in cyan circles) and approximate locations of actual features (in black stars) are also shown. 77
- 3.16 Average number of landmarks in the map. The number of pillars present in the experimental area was 36. It is clear that both RFS-CMSLAM solutions produce a closer estimate to the actual pillar count (number of actual landmarks), while RB-CMSLAM solution produces a larger estimate. 78

- 4.1 Proposed CMSLAMMOT solution. A particle filter is used to estimate the joint vehicle trajectories. A GM-PHD filter capable of handling switching multi-model motion is used to fuse the observations collected by individual collaborating robot and produce both a static feature map map and a dynamic feature map. The resultant static feature map is then used to estimate the joint vehicle trajectories of the collaborating vehicles, and the dynamic feature map is used in the navigation stack to avoid collisions with dynamic objects present in the sensor FOV. 88
- 4.2 How the PHD of the dynamic features evolves over time in an example CMSLAMMOT scenario. The yellow circles denote actual static landmark locations, while dotted lines denote the observations of each vehicle. A dynamic object emerges at (15,-2.5) at time step k and moves towards the vehicles. It can be seen that over time, the Gaussian correspond to the dynamic object increases in magnitude, while its spread is reduced. 98
- 4.3 Ground truth of the vehicle trajectories (vehicle 1 in blue and the vehicle 2 in black) with randomly placed static features superimposed on the set of accumulated false measurements (in green) recorded during a sample run under the clutter rate $\lambda_c^r = 1$, where $r = 1, 2$ denotes the vehicle number. The tracks of the dynamic features represent in cyan. The number of static features present in the environment is 57 and the number of dynamic features used is 5. 106

4.4	A comparison of estimated vehicle trajectories (first vehicle in green and second in red) superimposed on the ground truth (first vehicle in blue and second in black). Actual static feature locations are shown in black stars and the corresponding estimates are shown in cyan circles. Actual trajectories of the dynamic features are shown in yellow and the corresponding estimates are shown in magenta circles.	106
4.5	RMS Vehicle position errors from 15 MC runs (first vehicle in green and the second in red).	107
4.6	A comparison of average number of static features present in the map. It is clear that the proposed CMSLAMMOT algorithm produces almost identical results to that of the CMSLAM solution.	107
4.7	A comparison of average map error in terms of OSPA distance. It can be seen that the CMSLAMMOT algorithm produces slightly larger map errors compared to the CMSLAM algorithm.	108
4.8	A picture taken during the experiment. The pillars present in the picture were used as static features and two pedestrians were used as dynamic features present in the environment. This figure shows two pedestrians crossing the overlapped sensor FOVs at the same time. .	108
4.9	Estimated vehicle trajectories (first vehicle in green and the second in red) superimposed on the known way-points (first vehicle in blue and the second in red) along with estimated static feature locations (cyan). Estimated dynamic feature trajectories are shown in magenta and the corresponding actual trajectories (approximately) is shown in black.	110

4.10	Single vehicle RFS SLAM with MOT (SLAMMOT). A comparison of estimated vehicle trajectories (in red) and estimated moving target trajectories (blue circles) superimposed on ground truth of vehicle trajectories (in blue) and ground truth of moving target trajectories (in black) with estimated features (red circles) and actual features (green stars).	110
4.11	Single vehicle RFS SLAM with MOT. RMS Vehicle position error of 10 MC runs under low (in green) and high (in red) clutter conditions.	111
4.12	Average map error (static features) during RFS SLAMMOT in terms of OSPA distance under low (in green) and high (in red) clutter conditions.	111
4.13	Average number of static features present in the map during the SLAMMOT under low (in green) and high (in red) clutter conditions. The actual number of static features observed is shown in black	112
5.1	RFS Single-Vehicle SLAM based sub-mapping process with two vehicles. At points A and C each vehicle initiates the RFS Single-Vehicle SLAM process with a new local frame of reference and reach points B and D respectively. Local sub-maps produced, along with the sequence of control commands are communicated into the CMSLAM algorithm. This process repeats from points B and D. Black stars denote the observed landmarks during the process.	122
5.2	The HCMSLAM algorithm for a three vehicle scenario. Each participating vehicle produces a local sub-map and a vehicle trajectory, which are used as inputs in the upper level CMSLAM filter along with the sequence of applied control commands.	123

5.3	Proposed HCMSLAM solution. Each collaborating robot performs RFS-MONO-SLAM and produce a vehicle trajectory and a sub-map. The sub-map along with the applied control commands applied during the sub-mapping process are communicated in to the fusion node. At the fusion node, a hierarchical RFS CMSLAM algorithm is run, which fuses all such information from all the collaborative robots and produces a single global map and estimate the global vehicle trajectories of the collaborating robots.	126
5.4	Estimated vehicle trajectories (vehicle 1 in green and vehicle 2 in red) along with estimated feature locations (in cyan circles).	144
5.5	Estimated vehicle trajectories (vehicle 1 in green and vehicle 2 in red) along with estimated feature locations (in cyan circles).	144
5.6	RMS Vehicle position errors of 50 MC runs.	145
5.7	Average number of landmarks in the map, superimposed on the actual number of landmarks (in black).	145
5.8	Average map error in terms of OSPA distance.	146
5.9	Average simulation time taken to produce a sub-map (RFS mono-SLAM).	146
5.10	Average simulation time (seconds) taken to fuse sub-maps (RFS HCM-SLAM).	146

List of Acronyms

AV	Autonomous Vehicle
GPS	Global Positioning System
SLAM	Simultaneous Localization and Mapping
CMSLAM	Collaborative Multi-Vehicle SLAM
FOV	Field of View
EKF	Extended Kalman Filter
CLSF	Constrained Local Submap Filter
GMRF	Gaussian Markov Random Field
RFS	Random Finite Set
FISST	Finite Set Statistics
PHD	Probability Hypothesis Density
GM-PHD	Gaussian mixture Probability Hypothesis Density
RB-CMSLAM	Rao-Blackwellized Collaborative Multi-Vehicle SLAM
RFS-CMSLAM	Random Finite Set Collaborative Multi-Vehicle SLAM
OSPA	Optimal Subpattern Assignment
MOT	Moving Object Tracking
CMSLAMMOT	Collaborative Multi-Vehicle SLAM with MOT
HCMSLAM	Hierarchical Collaborative Multi-Vehicle SLAM

Nomenclature

k	Time step
M_k	Landmark map RFS
$U_k^{(r)}$	Control input to vehicle r
$U_{0:k}^{(r)}$	History of control inputs applied to vehicle r
$X_k^{(r)}$	Pose of vehicle r
$X_{0:k}^{(r)}$	Trajectory of vehicle r
$Z_k^{(r)}$	Measurement RFS of vehicle r
$Z_{1:k}^{(r)}$	History of measurements of vehicle r
$\Gamma_k(X_k^{(1)}, X_k^{(2)})$	RFS of new landmarks appearing in joint sensor FOV ($r = 1, 2$)
$\Upsilon(\zeta_{k-1})$	Bernoulli RFS of predicted state of landmark with state ζ_{k-1}
P_S	Probability of reappearance
$f_M(\zeta_k \zeta_{k-1}, \cdot)$	Landmark state transition density
$f_M(M_k M_{k-1}, \cdot)$	RFS map transition density
C_k	Clutter RFS
$\Theta_k^{(r)}(\zeta_k)$	Measurement RFS of landmark at ζ_k
P_D	Probability of detection
$g_k^{(r)}(z^{(r)} \zeta_k, X_k^{(r)})$	Measurement likelihood
$g_k^{(r)}(Z_k^{(r)} M_k, X_k^{(r)})$	Pdf of $Z_k^{(r)}$
l_k	Number of landmarks in the map
$p_{k k}(M_k \cdot)$	Landmark map posterior

$D_{k k}(\zeta_k \cdot)$	PHD of the landmark map
$D_{k k-1}(\zeta_k \cdot)$	PHD of the predicted landmark map
$b_k(\zeta_k X_k^{(1)}, X_k^{(2)})$	Intensity of the newly appearing landmarks
$\lambda_c^{(r)}$	Clutter rate of vehicle r
N_s	Number of particles
$f_X^{(r)}(X_k^{(r)} X_{k-1}^{(r),[i]}, U_k^{(r)})$	State transition density of vehicle r
$\eta_{k-1}^{[i,j]}$	Weight of the j th Gaussian component of the i th trajectory-conditioned PHD of map.
$\mu_{k-1}^{[i,j]}$	Mean of the j th Gaussian component of the i th trajectory-conditioned PHD of map.
$P_{k-1}^{[i,j]}$	Covariance of the j th Gaussian component of the i th trajectory-conditioned PHD of map.
$w^{[i]}$	Weight of i th sample trajectory
S_k	RFS of static feature map
D_k	RFS of dynamic features in the sensor FOV
$t_{k k-1}(\beta_k \beta_{k-1})$	Transition density between motion models
$ExtFoV(X_k^{(r)})$	Extended sensor FOV of vehicle r

Chapter 1

Introduction

1.1 Motivation

Autonomous Vehicles (AV) are making a significant impact in day-to-day applications, including, search and rescue [1] [2] [3], port and harbor surveillance [4] [5], hydrographic exploration [6], detection and tracking of moving objects [7] [8], intelligent transportation [9] to name a few. These tasks are not easily accomplished using a single AV due to the sheer complexity and extent of the environment in addition to the problem itself. As such, researchers are focusing more on using multiple AVs to collaboratively perform more complex missions. Having many AVs may not only provide additional performance gains in terms of speed, accuracy, reliability and robustness, but allow heterogeneity to be exploited to achieve better results.

For cost effective utilization in various missions, it is essential for AVs to localize themselves in possibly unknown and unstructured environments in which they are deployed. For example to perform a hydrographic survey in a shallow water marine environment, multiple autonomous surface vehicles (ASVs) [10] might be deployed together with autonomous underwater vehicles (AUVs) in order to collaboratively

scan the sea bed [6] [11] [12] using multi-beam sonar scanners for extracting various objects on the sea bed. The features that are extracted have to be combined with accurate positional information in order to build accurate maps to be used for further examination. Even though GPS data from surface vehicles can be used for localization, there is still a possibility that due to changing atmospheric and field conditions such as canopy tunnels etc., GPS measurements are either unavailable or unacceptably inaccurate.

We propose to overcome these problems through exploitation of artificial and/or naturally occurring features in the operating environment for collaborative mapping and localization of the vehicles. Using the well-known simultaneous localization and mapping (SLAM) [13] [14] framework as the basis, there exist few Collaborative Multi-Vehicle simultaneous localization and mapping (CMSLAM) algorithms in the literature. Although such algorithms perform adequately well in simulated environments, in field conditions they tend to perform poorly, because of high clutter returns from the environment and also inaccurate sensor and motion models. Out of these, clutter produced by sensors can significantly degrade navigation performance, because such data, if not processed correctly cause SLAM algorithms to produce inaccurate and inconsistent results. As an example multi-beam sonar sensor used in shallow sea-bed mapping applications [15] produces measurement data with high clutter near the overlapping areas of beam scans. This requires the sensor data be filtered from clutter before being used for feature extraction.

1.2 Thesis Objective and Approach

The main objective of this research is to develop algorithms to solve the Collaborative Multi-Vehicle Simultaneous Localization and Mapping (CMSLAM) problem

for a fleet of autonomous vehicles so as to improve the navigation in dynamic, uncertain and unstructured environments. More specifically this involves developing algorithms for a set of collaboratively working autonomous vehicles with a variety of sensors and actuators capable of gathering information under field conditions, transmitting information amongst each other and using such information to jointly localize and map the environment in a coordinated and a collaborative manner. The proposed solutions should overcome key practical issues such as non-linear motion models, measurements with false alarms (clutter), landmark detection uncertainty, landmarks survival uncertainty, exponentially increasing communication bandwidth requirements and computational power requirements.

1.3 Contributions

The key contributions of the thesis are summarized as follows,

- Two novel Collaborative Multi-Vehicle SLAM algorithms are proposed by representing the landmark map and measurements as random finite sets (RFSs) and formulating the full CMSLAM problem using finite set statistics (FISST). These solutions address non-linear motion models, map management, data association, landmark detection uncertainty, landmark survival uncertainty and measurement clutter removal in a single filtering step. The performance of the proposed solution is evaluated using simulation results and experimental results.
- A novel solution to the Collaborative Multi-Vehicle SLAM with Moving Object Tracking (CMSLAMMOT) problem is presented by extending the proposed RFS CMSLAM formulation. This algorithm can be used to solve a large array of practical CMSLAM problems under various dynamic environmental

conditions found in both military and civilian applications. Efficacy of the proposed solution is evaluated using simulation and experimental results.

- Finally, a computationally cheaper, hierarchical collaborative multi-vehicle SLAM solution (HCMSLAM) is presented by introducing a RFS mono-SLAM based sub-mapping process. Instead of communicating applied control and observation data at each time step, each vehicle is allowed to perform RFS mono-SLAM and the resultant sub-maps along with the applied control commands are used in a higher level HCMSLAM algorithm in order to approximate the full CMSLAM posterior. This solution reduces the communication bandwidth requirements and heavy computational load at the fusion node.

1.4 Organization of the thesis

Chapter 2 presents a brief introduction into the SLAM problem and provides an overview of most commonly used SLAM algorithms such as extended Kalman filter based SLAM (EKF-SLAM) and particle filter based SLAM (Fast SLAM) solutions. The recently developed random finite set (RFS) formulation for SLAM approach is also summarized. Then the state of the art solutions for Multi-Vehicle SLAM problem and their pros and cons are presented. Chapter 3 presents a novel solution to the collaborative multi-vehicle SLAM (CMSLAM) problem by representing the landmark map and the measurements as RFSs. The efficacy of the proposed algorithm is presented via simulations and experiments using two ground robots. In Chapter 4 a further improvement to the proposed CMSLAM solution is made by introducing moving target tracking into the SLAM problem by solving the collaborative multi-vehicle SLAM algorithm with moving target tracking (CMSLAMMTT). The effectiveness of the proposed algorithm is presented using simulations and ex-

periments using ground robots. Chapter 5 presents a hierarchical solution based on the RFS single vehicle SLAM to solve the CMSLAM problem in order to address the issue of exponentially increasing communication bandwidth requirements. The performance of this algorithm is evaluated using simulations and experiments using ground robots. Finally, Chapter 6 presents the conclusion of the thesis together with major contributions.

Chapter 2

Literature Review

In this Chapter we present an overview of state of the art SLAM solutions. In section 2.1, a brief introduction to the SLAM problem is presented. Section 2.2 presents an overview of the state of the art mono-SLAM solutions. Most commonly used SLAM algorithms are outlined and their properties are also discussed. In section 2.3, Collaborative Multi-Vehicle SLAM (CMSLAM) problem is formulated. The state of the art CMSLAM solutions and their limitations are also discussed. Section 2.4 concludes the chapter by proposing alternative algorithms to the CMSLAM problem in order to address limitations exist in the current state of the art solutions.

2.1 The SLAM problem

The simultaneous localization and mapping (SLAM) [13] [14] problem examines whether a mobile robot placed in an unknown environment can incrementally build a map of its surrounding and simultaneously localize itself with respect to this map. Both the vehicle trajectory of the platform and the locations of all landmarks are estimated online without the knowledge of initial vehicle state. A robot uses its

proprioceptive sensors such as gyroscope, IMU, accelerometer etc. in order to predict its state during the motion. It observes the environment using its exteroceptive sensor which is usually a laser range finder, a radar or a sonar scanner. Solving the SLAM problem is considered as one of the key prerequisite in order to build truly autonomous mobile robots.

2.2 The mono-SLAM problem

Based on the seminal work of Smith et al. [16] [17], the very first classical solution to the Single-Vehicle SLAM (Herein after referred to as mono-SLAM) problem, the EKF-SLAM algorithm was proposed by Dissanayake et al. in [18]. The EKF-SLAM solution represents the landmark map and the set of observations as vectors, and the observation model and vehicular motion model are represented in state space form with additive zero mean Gaussian observation and motion noise. A joint posterior density consist of the vehicle state and the set of observed landmarks, given the recorded observations, control inputs and the initial vehicle position is propagated forward in time to solve the SLAM problem.

2.2.1 Problem formulation

Let $U_{1:k} = [U_1, U_2, \dots, U_k]^T$ denote the time sequence of control commands applied to the vehicle up to time k , where U_k denotes the control command applied at time k . Let the time sequence of poses history of the vehicle be denoted by $X_{1:k} = [X_1, X_2, \dots, X_k]^T$, where X_k denotes the pose of the vehicle with respect to the global reference frame, at time k . Let the time sequence of sets of measurements obtained using exteroceptive sensors mounted on each vehicle be denoted by $Z_{1:k} = [Z_1, Z_2, \dots, Z_k]$, where $Z_k = [z_{k,1}, z_{k,2}, \dots, z_{k,n_k}]$ denotes the measurement set

received at time k , while n_k denotes the number of measurements. Let the map at time k be denoted by the set $M_k = [m_{k,1}, m_{k,2}, \dots, m_{k,l_k}]$, where l_k denotes the number of landmarks available in the map at time k .

Using this information, the two major formulations of the classical SLAM problem, the online SLAM algorithm and the full SLAM algorithm [19] can be outlined as follows.

The online SLAM algorithm

In the online SLAM algorithm was first introduced in the EKF-SLAM solution [18] [19], where we are required to evaluate the posterior probability density,

$$p_{k|k}(X_k, M_k | Z_{1:k}, U_{1:k}, X_0) \quad (2.1)$$

where X_0 is the initial vehicle position. The posterior distribution (2.1) is solved using a two-step recursive prediction and measurement update procedure given by,

$$p(X_k, M_k | Z_{1:k}, U_{1:k}, X_0) = \int p(X_k | X_{k-1}, U_k) p(X_{k-1}, M_k | Z_{1:k-1}, U_{1:k-1}, X_0) dX_{k-1} \quad (2.2)$$

and

$$p(X_k, M_k | Z_{1:k}, U_{1:k}, X_0) = \frac{p(Z_k | X_k, M_k) p(X_k, M_k | Z_{1:k-1}, U_{1:k}, X_0)}{p(Z_k | Z_{1:k-1}, U_{1:k})} \quad (2.3)$$

where equation (2.2) is the time update where the joint state of the map and the vehicle position is predicted using the SLAM posterior at time $k-1$ and the motion model of the vehicle given by,

$$p(X_k | X_{k-1}, U_k) \quad (2.4)$$

The vehicle state transition is assumed to be a Markov process, which the state X_k depends only on the previous vehicle state X_{k-1} and the control commands U_k applied at time k . The prediction is independent of the observations and the map. The equation (2.3) represent the measurement update step where the observation model is given by,

$$p(Z_k|X_k, M_k) \quad (2.5)$$

which describes the probability of producing an observation Z_k , given the landmark map M_k and the current vehicle position X_k . The denominator is the normalization constant which is obtained by,

$$p(Z_k|Z_{1:k-1}) = \int p(Z_k|X_k, U_{1:k})p(X_k|Z_{1:k-1}, U_{1:k}) \quad (2.6)$$

The full SLAM algorithm

The full SLAM solution was first introduced by Montemerlo et al. [20] [19] in their FastSLAM solution. In full SLAM, we are required to evaluate the posterior probability density,

$$p_{k|k}(X_{1:k}, M_k|Z_{1:k}, U_{1:k}, X_0) \quad (2.7)$$

where, instead of estimating the current vehicle position, the joint posterior consist of entire vehicle trajectory and the landmark map is estimated using the sequence of applied control commands, sequence of observations and the initial vehicle position.

2.2.2 State of the art mono-SLAM solutions

Based on this paradigm, the very first successful solution to the SLAM problem, the extended Kalman filter SLAM (EKF-SLAM) [18] was proposed, which gained an immense popularity among robotics researchers. Although EKF-SLAM lead to

a significant progress in the area by opening up new approaches, the work followed had to retain its original linear Gaussian assumptions, limiting the possible applications of SLAM. A consistency and convergence analysis of EKF-SLAM solution can be found in [21]. Montemarlo et al. [20] came up with the next notable conceptual improvement in the area, the FastSLAM algorithm. By introducing recursive Monte Carlo Sampling (particle filtering) and careful factorization of the full SLAM posterior into a product of the vehicle trajectory posterior and landmark map posterior conditioned on vehicle trajectory, FastSLAM catered non-linear process models and non-Gaussian pose distribution, making it a much suitable candidate algorithm for various field conditions. In addition to EKF-SLAM and FastSLAM, which are the most common SLAM algorithms in the literature, Thrun et al. proposed a new SLAM algorithm, that relies on the dual of the EKF, the extended information filter (EIF). This new SLAM algorithm, sparse extended information filter SLAM (SEIF-SLAM) [22], produces comparable results to the EKF-SLAM in terms of accuracy while having a much improved computational speed.

Based on these conventional SLAM solutions, a larger number of variant SLAM solutions can be found in the mobile robotics literature. In [23] [24] Williams et al. proposed a scalable SLAM solution for large scale mapping by introducing the concept of constrained local sub-map filtering (CLSF). In [25] Pinies et al. proposed a conditionally independent map fusion approach for SLAM in order to perform large scale map building. In [26], Pinies et al. proposed a graph based solution for the large scale SLAM problem. In [27], Martinez et al. proposed an unscented Kalman filter (UKF) [19] based SLAM solution for large scale SLAM in outdoor environments. In [28] Estrada et al. presented a hierarchical SLAM solution for real-time accurate mapping approach large outdoor environments. In [29] Wang et al. presented an efficient decoupled SLAM solution based on EIF. The salient feature of this approach is that the mapping problem is solved as a non-linear static

estimation problem and the localization problem to is solved as a low-dimensional estimation problem. More recently, in [30], Dellaert et al. presented an efficient solution for large scale SLAM problems based preconditioned conjugate gradients (PCG) approach [31].

In [32] Dong et al. introduced an efficient SLAM solution by combining a genetic algorithm for Rao-Blackwellised particle filtering (RBGAF-SLAM). In [33] Dong et al. extended their original SLAM solution in order cater for dynamic environments using raw sensor measurements. Moreover in [34], Dong et al. provided a detailed description of how to optimize their original RBGAF-SLAM parameters through genetic learning. Extending the EKF-SLAM solution, in [35] Paz et al. introduced a divide and conquer solution for SLAM in order to achieve faster measurement updates. In [36], Ni et al. proposed a sub-map based smoothing and mapping (SAM) [37] approach as a computationally efficient alternative to the conventional filtering based SLAM solutions. It is shown that the proposed solution drastically improves the running time performance while maintaining the global accuracy in very large scale environments.

Out of these, one of the most notable extensions to the existing conventional SLAM solutions is the introduction of Detection and Tracking of Moving Objects (DATMO) [38]. In [39] Wang et al. proposed the first practical solution to combine both SLAM and DATMO, where both SLAM problem and DATMO are solved simultaneously. In [40], Wang et al. extended their earlier work by presenting a theoretical framework for the SLAM with DATMO problem and demonstrated its performance using an EKF based implementation. Their work was based on decomposing the SLAM with DATMO problem into two separate estimators, namely, the SLAM posterior and the DATMO posterior and propagating them simultaneously over time. Even though this approach can be used in some applications, it has a few drawbacks.

Among them, decomposing the measurements into stationary objects and moving objects, inapplicability of complex multiple motion models per moving object are most notable. In order to classify measurements as stationary and dynamic, it is required to analyze the data over a sufficient period of time, using complex feature extraction algorithms. This may be undesirable in most practical applications if the results are to be used for making faster navigation decisions. In addition, most of the moving features such as pedestrians, animals or other vehicles may not have a single motion model, their motion models can be a mixture of motion models depending on the situation. Hence, the possibility of having multiple motion models per moving feature have to be taken into account. In addition, there are few more attempts to solve the SLAM with DATMO in mobile robotics literature. In [41] Ming et al. attempted a SLAM solution by combining both SLAM and detection and tracking of moving objects (DATMO) for dynamic environments. Huijing et al. proposed a SLAM solution for a dynamic large outdoor environments in [42]. In [43] Holz et al. proposed a continuous 3D sensing solution for navigation and SLAM under measurement clutter for dynamic unstructured environments.

In almost all conventional SLAM algorithms such as EKF-SLAM, FastSLAM or SEIF-SLAM and their derivatives, a posterior probability density of a vector consisting of the landmark map augmented with vehicle state is computed in order to solve the SLAM problem. All such algorithms require solving certain additional sub-problems outside the Bayesian SLAM recursion including data association, clutter filtering and map management in order to yield a consistent solution. Moreover, these conventional solutions assume several key simplifications that limit their practical applicability. The most notable assumptions are clutter free measurements, ideal landmark detection and data association.

Parallel to the developments in SLAM, are the advances in multi-target tracking

stemming from work by Mahler [44] [45], on optimal Bayesian multi-target filtering by representing the multi-target state by a finite set. This representation is fundamental as it provides a mathematically consistent notion of estimation error for multiple targets. Mahler devised a set of mathematical tools for characterizing random finite set (RFS) termed as finite set statistics (FISST) based on a notion of integration and density that is consistent with point process theory [46]. The RFS formulations can cater for detection uncertainty, data association uncertainty and false alarms (measurement clutter) that are commonly seen in real world target tracking applications.

The optimal Bayesian multi-target filter is mathematically sound, but computationally intractable in general, although recent results on an RFS conjugate prior [47] promises significant computational savings. As an alternative tractable solution, Mahler proposed to propagate the first-order moment of the optimal Bayesian filter [48], the probability hypothesis density (PHD), instead of the full posterior. PHD filter gained an immense popularity among the target tracking community with tractable sequential Monte Carlo (SMC) [46] [49] [50] and Gaussian mixture (GM) [51] implementations. Their convergence properties are discussed in [52] and [53] respectively. A survey on the implementation of PHD and CPHD filters can be found in [54]. In addition, in [55], Mahler proposed a generalization of the PHD recursion, the cardinalized PHD (CPHD) filter, which is also gaining a widespread appreciation among target tracking community, with its analytic implementation by Vo et al. [56].

Random finite set based tracking solutions are gaining a widespread appreciation among the target tracking community, resulting an increasing number of applications over the last few years. In [57], Sidenbladh et al. proposed a particle filter approximation for the optimal multi-target tracker for tracking an unknown and

varying number of vehicles in terrain. In [58] [59] [60], Clark et al. presented PHD filter based multi-target tracking scenarios for sonar images. Kalyan et al. [61] devised a particle based PHD filter for tracking in underwater sonar images. In [62] [63] Lee et al. presented a PHD filter based multi-target tracking solution for 3D Lidar data (Velodyne). In [64] Wood et al. presented a PHD filter based multi-target tracking scenario for microscopic video data. Masnadi-Shirazi et al. [65] presented a novel solution for tracking and separation of unknown time-varying number of sound sources in a reverberant environment. In [66] [67] [68], Tobias et al. applied PHD filters in passive coherent localization of RF signals. In addition, Wang et al. [69] have applied a PHD filter for tracking groups of humans in a digital video, while in [70] Ikoma et al. have applied a PHD filter for tracking feature points in time-varying optical images.

The overlap between multi-target tracking and SLAM naturally led to the adoption of RFS modeling in SLAM. The very first SLAM formulation using RFSs was presented by Mullane et al. in [71], while a Rao-Blackwellized PHD filtering solution (RFS SLAM) was proposed in [72] [73] by factorizing the full SLAM posterior into a product of path posterior and map posterior conditioned on vehicle path. The path posterior was propagated using a particle filter, while the map posterior was propagated using a GM implementation of a PHD filter. Most notably, by exploiting the structure of the map posterior and its representation, a new mathematically sound weighting function was introduced to the particle filter for path posterior estimation. Section 2.2.2 briefly summarizes the RFS representation of the landmark map and outlines the RFS SLAM solution. For a complete RFS Bayesian formulation and PHD filtering solution for SLAM we refer the reader to [73] [74]. More recently in [75], Lee et al. further improved the Mullane's RFS SLAM solution by considering a single-cluster multi-object model, where the state consists of landmarks conditioned on the vehicle state. It is claimed that the proposed solution indicates

superior landmark map estimation and localization performance compared to the original RFS SLAM solution.

In the remainder of the section, we summarize three most notable SLAM solutions up to date, EKF-SLAM, FastSLAM and the recently proposed RFS SLAM.

The EKF-SLAM algorithm

In the EKF-SLAM solution [18], vehicle motion is described as,

$$p(X_k|X_{k-1}, U_k) \Leftrightarrow X_k = f(X_{k-1}, U_k) + w_k \quad (2.8)$$

where $f(X_{k-1}, U_k)$ models the vehicle dynamics and w_k denotes the zero mean Gaussian control noise with covariance Q_k . The observation model can be written as,

$$p(Z_k|X_k, M_k) \Leftrightarrow Z_k = h(X_k, M_k) + v_k \quad (2.9)$$

where $h(X_k, M_k)$ denotes the measurement likelihood function and v_k denotes the zero mean Gaussian measurement noise with covariance R_k . Using this information, time update of the mean of the state vector can be obtained as,

$$\hat{X}_{k|k-1} = f(\hat{X}_{k-1|k-1}, U_k) \quad (2.10)$$

and the corresponding covariance matrix can be obtained as,

$$\hat{P}_{xx,k|k-1} = \nabla f P_{xx,k-1|k-1} \nabla f^T + Q_k \quad (2.11)$$

The mean of the measurement updated state vector can be obtained as follows,

$$\begin{bmatrix} \hat{X}_k \\ \hat{M}_k \end{bmatrix} = E \begin{bmatrix} \hat{X}_{k|k-1} \\ \hat{M}_{k-1} \end{bmatrix} + W_k \left[Z_k - h(\hat{X}_{k|k-1}, \hat{M}_{k-1}) \right] \quad (2.12)$$

and the corresponding covariance can be obtained as,

$$P_{k|k} = P_{k|k-1} - W_k S_k W_k^T \quad (2.13)$$

where

$$S_k = \nabla h P_{k|k-1} \nabla h^T + R_k \quad (2.14)$$

and

$$W_k = P_{k|k-1} \nabla h^T S_k^{-1} \quad (2.15)$$

The FastSLAM algorithm

The FastSLAM algorithm [20], which is based on recursive Monte Carlo sampling or particle filtering [76], caters non-linear process model and non-Gaussian pose distribution. The joint density of the vehicle pose and the landmark map in the full SLAM posterior (2.7) is factorized into the vehicle trajectories posterior and the landmark map posterior conditioned on the landmark map as follows.

$$p(X_{0:k}, M_k | Z_{1:k}, U_{1:k}, X_0) = p(X_{0:k} | Z_{1:k}, U_{1:k}, X_0) p(M_k | X_{0:k}, Z_{1:k}) \quad (2.16)$$

The distribution of joint history of vehicle poses and the landmark map is represented by a set of N weighted particles as follows.

$$\{w_k^{(i)}, X_{0:k}^{(i)}, p(M_k | X_{0:k}^{(i)}, Z_{1:k})\}_{i=1}^N \quad (2.17)$$

where the map accompanying each particle is composed of independent Gaussian distributions given by,

$$p(M_k | X_{0:k}^{(i)}, Z_{1:k}) = \prod_j^{l_k} p(m_{k,j} | X_{0:k}^{(i)}, Z_{1:k}) \quad (2.18)$$

Assuming that the joint landmark map and the vehicle trajectory at time $k - 1$ can be represented by $\{w_{k-1}^{(i)}, X_{0:k-1}^{(i)}, p(M_{k-1} | X_{0:k-1}^{(i)}, Z_{1:k-1})\}_i^N$, the general form of the Rao-Blackwellised particle filter can be outlined as follows,

1. A new particle is drawn according to the motion model,

$$X_k^{(i)} \sim p(X_k | X_{k-1}^{(i)}, U_k) \quad (2.19)$$

and this new particle is added to the particle history $X^{(i)} \triangleq \{X_{0:k-1}^{(i)}, X_k^{(i)}\}$

2. Now, particles are weighted according to the importance function as follows,

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(Z_k | X_{0:k}^{(i)}, Z_{1:k-1}) p(X_k^{(i)} | X_{k-1}^{(i)}, U_k)}{\pi(X_k^{(i)} | X_{0:k-1}^{(i)}, Z_{1:k}, U_k)} \quad (2.20)$$

3. Re-sampling is performed by selecting particles according to their weights. Each new particle is given the uniform weight of $w_k^{(i)} = 1/N$.
4. Now for each particle, an EKF update is performed on the observed landmarks with known vehicle position.

The RFS SLAM algorithm

This section summarizes the RFS representation of the landmark map and measurement model, and outlines the Bayesian formulation of the RFS SLAM solution under data association uncertainty and landmark detection uncertainty.

RFS representation of the landmark map The RFS representation caters for a consistent error matrix in multi-target tracking, which is an essential requirement in order to evaluate the quality of the estimation process. Similarly, in the SLAM literature, although the localization error is directly associated with the mapping error, a measure of the quality of the map has not been proposed. In [73] Mullane et al. argues that by representing the landmark map as a RFS, it is possible to adequately measure the quality of the map despite the ordering and miss-detections of the landmarks. For completeness, we will draw the same example as presented in [73] to illustrate this as follows.

Let's consider a vector based single vehicle SLAM scenario with two actual landmarks at (0,0) and (1,1). Based on this, the actual landmark map can be represented by the vector $M = [0 \ 0 \ 1 \ 1]^T$. Since the estimated landmark locations are stacked in to the map depending on the order of observations (based on the trajectory taken), estimate of the landmark map may result in the vector $\hat{M} = [1 \ 1 \ 0 \ 0]^T$. Now, although we have an exact estimate of the map, still we can obtain the map error in terms of Euclidean distance as $\|M - \hat{M}\| = 2$. But instead, if the ordering of the landmarks is changed, we have an exact estimate of the landmark map, with a resulting a map error of 0. This means that, the estimated map representation may contain all possible permutations of the elements of an actual landmark map represented by a vector. Hence it is apparent that, only a set based representation of the estimated landmark map allows such a comparison of the estimated landmark map with a true map.

Another complication arises in conventional SLAM approaches in the case of missed detections. Let's say instead of both landmarks, the estimated map contains a single landmark, say at (0,0). In this case, it is impossible to evaluate the mapping error as there is no method that can compare the two vectors having different number

of elements. In order to address such issues, Mullane et al. [73] devised the RFS SLAM solution, in which the landmark map is represented as a RFS. Since the error between two sets is a well-defined concept (for example, OSPA distance [77]), such a representation provides a consistent estimation error of the landmark map.

Map management Map management in RFS SLAM solution can be described using the RFS map transition model and the RFS measurement model as follows.

Map transition model The new landmarks appearing in the sensor FOV of vehicle at time k is modeled as the RFS $\Gamma_k(X_k)$. Let the RFS of map at time $k - 1$ be denoted by M_{k-1} , then the map transition model at time k is given by,

$$M_k = \Gamma_k(X_k) \cup \left[\bigcup_{\zeta_{k-1} \in M_{k-1}} \Upsilon(\zeta_{k-1}) \right] \quad (2.21)$$

where, the Bernoulli RFS $\Upsilon(\zeta_{k-1})$ denotes the predicted state of the landmark $\zeta_{k-1} \in M_{k-1}$, which takes the form $\{\zeta_k\}$ with probability of reappearance $P_S(\zeta_{k-1})$ or $\emptyset$ with probability $1 - P_S(\zeta_{k-1})$. The probability density of state transition of the landmark from ζ_{k-1} to ζ_k conditioned on the probability of survival is given by $f_M(\zeta_k | \zeta_{k-1}, X_k)$. Note that, $\Gamma_k(X_k)$ is independent of the landmarks that are reappearing from the map M_{k-1} . The corresponding RFS map transition density [73] can be written as,

$$f_M(M_k | M_{k-1}, X_k) = \sum_{W \subseteq M_k} f_M(W | M_{k-1}) f_\Gamma(M_k - W | X_k) \quad (2.22)$$

where, $f_M(W | M_{k-1})$ is the transition density of the set of features existing in the map, M_{k-1} from time $k - 1$ to time k , while $f_\Gamma(M_k - W | X_k)$ represents the density of the RFS $\Gamma_k(X_k)$ of new landmarks appearing in the joint sensor FOV at time k .

Measurement Model Let M_k denote the landmark map, while C_k denote the RFS representing clutter received from exteroceptive sensor mounted on the vehicle at time k , then corresponding measurements can be represented by the RFS,

$$Z_k = C_k \cup \left[\bigcup_{\zeta_k \in M_k} \Theta_k(\zeta_k) \right] \quad (2.23)$$

where $\Theta_k(\zeta_k)$ is a Bernoulli RFS representing the measurement corresponds to the landmark $\zeta_k \in M_k$. Due to the limited FOV of the sensor, $\Theta_k(\zeta_k)$ can have a value of the form $\{z_k\}$ with probability of detection $P_D(\zeta_k|X_k)$ or $\emptyset$ with a probability of $1 - P_D(\zeta_k|X_k)$. Note that the probability of detection, is a function of landmark state and vehicle position. Conditioned on detection, measurement likelihood function corresponds to the observation of landmark with state ζ_k is given by $g_k(z|\zeta_k, X_k)$. The corresponding RFS measurement density can be written as,

$$g_k(Z_k|M_k, X_k) = \sum_{W \subseteq Z_k} g_Z(W|M_k, X_k) g_C(Z_k - W) \quad (2.24)$$

where, $g_Z(W|M_k, X_k)$ denotes the density of the feature observations, while $g_C(Z_k - W)$ denotes the density of clutter.

Bayesian RFS SLAM filter Let the time sequence of sets of measurements obtained using exteroceptive sensors mounted on the vehicle be denoted by $Z_{1:k} = [Z_1, Z_2, \dots, Z_k]$, where $Z_k = \{z_{k,1}, z_{k,2}, \dots, z_{k,n_k}\}$ denotes the measurement set received at time k , while n_k denotes the number of measurements. Let the map at time k be denoted by the set $M_k = \{m_{k,1}, m_{k,2}, \dots, m_{k,l_k}\}$, where l_k denotes the number of landmarks available in the map at time k .

Using this information the RFS SLAM posterior can be written as follows,

$$p(X_{1:k}, M_k | Z_{1:k}, U_{1:k}, X_0) \quad (2.25)$$

where, we are required to jointly propagate the landmark map and the history of vehicle pose, using observations and control inputs and initial vehicle position. The SLAM posterior encapsulates all statistical information about the landmark map and the vehicle trajectories given the initial vehicle pose, history of control inputs and the history of observations up to time k .

Although RFS SLAM posterior incorporates all the statistical information relevant to the SLAM problem, it is impossible to propagate it as it is in time since the RFS SLAM posterior is numerically intractable. In order to address this, similar to FastSLAM, the RFS SLAM posterior is factorized into the vehicle trajectory posterior and the landmark map posterior conditioned on the vehicle trajectory [72] [73]. The vehicle trajectory is propagated using a Rao-Blackwellized particle filter and the landmark map is propagated using a PHD filter. This caters propagation of the RFS SLAM posterior in a Bayesian filtering framework with inbuilt map management, data association and clutter filtering while catering for landmark map detection uncertainty and landmark survival uncertainty.

It follows that,

$$p(X_{1:k}, M_k | Z_{1:k}, U_{1:k}, X_0) = p(X_{1:k} | Z_{1:k}, U_{1:k}, X_0) \times p(M_k | X_{0:k}, Z_{1:k}) \quad (2.26)$$

This decomposition allows propagation of the landmark map and the vehicle trajectory as two separate estimation problems. The vehicle trajectory is propagated using a RB particle filter and the landmark map is propagated using a GM implementation of a PHD filter. Since the landmark map is conditioned on the vehicle

trajectory, for each particle in the particle filter, a separate landmark map is propagated in time.

The vehicle trajectory estimation The vehicle trajectory posterior given by,

$$p(X_{1:k}|Z_{1:k}, U_{1:k}, X_0) = g_k(Z_k|Z_{0:k-1}, X_{0:k}) \times \frac{f_X(X_k|X_{k-1}, U_k)p_{k-1}(X_{k-1}|Z_{0:k-1}, U_{0:k-1}, X_0)}{g_k(Z_k|Z_{0:k-1})} \quad (2.27)$$

is propagated using a RB particle filter. In (1.27), $g_k(Z_k|Z_{0:k-1})$ is a normalization constant.

The landmark map posterior estimation The landmark map posterior can be decomposed into a recursive prediction/update procedure as follows,

$$p_{k|k-1}(M_k|Z_{1:k-1}, X_{0:k}) = \int f_M(M_k|M_{k-1}, X_k)p_{k-1}(M_{k-1}|Z_{1:k-1}, X_{0:k-1})\delta M_{k-1} \quad (2.28)$$

$$p_{k|k}(M_k|Z_{1:k}, X_{0:k}) = g_k(Z_k|M_k, X_k) \times \frac{p_{k|k-1}(M_k|Z_{1:k-1}, X_{0:k})}{l_{k|k-1}(Z_k|Z_{1:k-1}, X_{0:k})} \quad (2.29)$$

where the denominator in (1.29) is a normalization constant. The map posterior is modeled using FISST and propagated using a Gaussian mixture implementation of a PHD filter [73]. The PHD filter propagates the intensity of the landmark map RFS, from which it is possible to obtain the number of landmarks and the landmark positions.

Assuming that the number of false measurements (clutter) produced by the exteroceptive sensor mounted on the vehicle is Poisson distributed with an average of λ_c , and their physical distribution is given by $c_k(z_k)$, then PHD of the landmark map can be obtained using a Gaussian mixture (GM) implementation of a PHD filter [72] [73] as follows.

The predicted PHD is given by,

$$D_{k|k-1}(\zeta_k|X_{0:k}, Z_{1:k-1}) = b(\zeta_k|X_k) + P_S(\zeta_k)D_{k-1}(\zeta_k|X_{0:k-1}, Z_{1:k-1}) \quad (2.30)$$

where, $b(\zeta_k|X_k)$ denotes the intensity of the newly appearing landmarks. The measurement updated PHD is given by,

$$D_k(\zeta_k|X_{0:k}, Z_{1:k}) = [1 - P_D(\zeta_k|X_k)]D_{k|k-1}(\zeta_k|X_{0:k}, Z_{1:k-1}) + \sum_{z \in Z_k} \frac{P_D(\zeta_k|X_k)g_k(z|\zeta_k, Z_k)D_{k|k-1}(\zeta_k|X_{0:k}, Z_{1:k-1})}{\lambda_c c_k(z|X_k) + \int P_D(\zeta_k|X_k)g_k(z|\zeta_k, Z_k)D_{k|k-1}(\zeta_k|X_{0:k}, Z_{1:k-1})d\zeta_k} \quad (2.31)$$

where $\zeta_k \in M_k$, and $P_D(\zeta_k|X_k)$ denotes the probability of detection, while $P_S(\zeta_k)$ denotes the probability of survival of a landmark from time step $k - 1$ to k . Both probabilities are assumed to be constants in general.

RFS SLAM posterior estimation Now, the RFS SLAM posterior (1.25) can be represented using a set of N particles as follows,

$$\{w_k^{[i]}, X_{0:k}^{[i]}, D_k^{[i]}(\zeta_k|X_{0:k}^{[i]}, Z_{1:k})\}_{i=1}^N \quad (2.32)$$

where, $D_k^{[i]}(\zeta_k|X_{0:k}^{[i]}, Z_{1:k})$ denotes the PHD of the landmark map posterior conditioned on the vehicle trajectory. The vehicle trajectory and the PHD of the particle with the highest weight is considered as the best representation of the estimate of the vehicle trajectory and the landmark map given the sequence of applied control inputs, history of the sets of observations and the initial vehicle position.

Let the PHD of the landmark map corresponds to the i th particle at time $k - 1$ be

denoted by a mixture of Gaussians of the form,

$$D_{k-1|k-1}(\zeta_{k-1}|Z_{1:k-1}, X_{0:k-1}^{[i]}) = \sum_{j=1}^{J_{k-1}^{[i]}} \eta_{k-1}^{[i,j]} \mathcal{N}(\zeta_{k-1}; \mu_{k-1}^{[i,j]}, P_{k-1}^{[i,j]}) \quad (2.33)$$

where, J_{k-1} is the number of Gaussians, while $\eta_{k-1}^{[i,j]}$, $\mu_{k-1}^{[i,j]}$ and $P_{k-1}^{[i,j]}$ respectively denote weight, mean and covariance of the j^{th} Gaussian. Let the intensity of the new landmarks appearing in the sensor FOV of vehicle at time k is represented as a mixture of Gaussians as follows,

$$b_k(\zeta_k|X_k^{[i]}) = \sum_{j=1}^{J_{b,k}} \eta_{b,k}^{[i,j]} \mathcal{N}(\zeta_k; \mu_{b,k}^{[i,j]}, P_{b,k}^{[i,j]}) \quad (2.34)$$

where, $J_{b,k}$ denotes the expected number of new landmarks appearing in the sensor FOV of the vehicle. Then the PHD update step results in a mixture of Gaussians as shown in Table 2.1. A pruning procedure (Table.2.2) is carried out in order to reduce the number of Gaussian components and merge Gaussians with highest weights.

Data association, clutter filtering and map management in RFS SLAM

solution Let's take an example in order to illustrate how RFS SLAM solution performs clutter filtering and map management, avoiding explicit data association. The Fig.2.1 shows an example how RFS SLAM solution works in practice. Here we have extracted a scenario where, a single vehicle performs RFS SLAM from time step $k-1$ to $k+2$. At time $k-1$, the vehicle observes four actual features and two false features from $(-6,10)$ and $(-4,4)$ (Fig.3.2(a)), this results in the updated PHD of the map as shown in Fig.3.2(a). The two Gaussians at $(-6,10)$ and $(-4,4)$ corresponds to the false features observed. Then, at time k , the vehicle further moves forward observing the same actual features, while observing two new false features at $(-5,0)$ and $(-3,1)$ (Fig.3.2(c)). Corresponding PHD of the map is shown in Fig.3.2(d), where

Table 2.1: PHD Filter

```

1. Given:  $\{w_{k-1}^{[i]}, X_{0:k-1}^{[i]}, D_{k-1}^{[i]}(\zeta_{k-1}|X_{0:k-1}^{[i]}, Z_{1:k-1})\}_{i=1}^N$ 
2.  $l = 0$ 
3. for  $j = 1$  to  $J_{b,k}$  (Newly appearing Gaussians)
4.    $l = l + 1$ 
5.    $\eta_{k|k-1}^{[i,l]} = P_S * \eta_{b,k}^{[i,j]}, \mu_{k|k-1}^{[i,l]} = \mu_{b,k}^{[i,j]}, P_{k|k-1}^{[i,l]} = P_{b,k}^{[i,j]}$ 
6. end
7. for  $j = 1$  to  $J_{k-1}^{[i]}$  (Prediction of existing Gaussians)
8.    $l = l + 1$ 
9.    $\eta_{k|k-1}^{[i,l]} = \eta_{k-1}^{[i,j]}, \mu_{k|k-1}^{[i,l]} = \mu_{k-1}^{[i,j]}, P_{k|k-1}^{[i,l]} = P_{k-1}^{[i,j]}$ 
10. end
11.  $J_{k|k-1}^{[i]} = l$ 
12. for  $i = 1$  to  $N$ 
13.   for  $j = 1$  to  $J_{k|k-1}^{[i]}$  (Preparing for update)
14.      $\hat{z}^{[i,j]} = h_k(\mu_{k|k-1}^{[i,j]}, X_k^{[i]}, 0)$ 
15.      $S_k^{[i,j]} = U_k^{[i,j]} R_k^{[i,j]} U_k^{[i,j]T}$ 
16.      $K_k^{[i,j]} = P_{k|k-1}^{[i,j]} H_k^{[i,j]} (S_k^{[i,j]})^{-1}$ 
17.      $P_{k|k}^{[i,j]} = (I - K_k^{[i,j]} H_k^{[i,j]}) P_{k|k-1}^{[i,j]}$ 
18.   end
19.   for  $j = 1$  to  $J_{k|k-1}^{[i]}$  (Missed detections)
20.      $\eta_k^{[i,j]} = (1 - P_D) \eta_{k|k-1}^{[i,j]}$ 
21.      $\mu_k^{[i,j]} = \mu_{k|k-1}^{[i,j]}$ 
22.      $P_k^{[i,j]} = P_{k|k-1}^{[i,j]}$ 
23.   end
24.    $l = 0$ 
25.   for each  $z \in Z_k$  (Detections)
26.      $l = l + 1$ 
27.     for  $j = 1$  to  $J_{k|k-1}^{[i]}$ 
28.        $\tau^{(j)} = P_D \eta_k^{[i,j]} N(z_k; h_k(\mu_{k|k-1}^{[i,j]}, X_k^{[i]}, 0), S_k^{[i,j]})$ 
29.        $\mu_k^{[i, lJ_k^{[i]}+j]} = \mu_{k|k-1}^{[i,j]} + K_k^{[i,j]}(z - \hat{z}^{[i,j]})$ 
30.        $P_k^{[i, lJ_k^{[i]}+j]} = P_k^{[i,j]}$ 
31.     end
32.     for  $j = 1$  to  $J_{k|k-1}^{[i]}$ 
33.        $\eta_k^{[i, lJ_k^{[i]}+j]} = \tau^{(j)} / c(z) + \sum_{m=1}^{J_{k|k-1}^{[i]}} \tau^{(m)}$ 
34.     end
35.   end
36.    $J_k^{[i]} = (l + 1) J_{k|k-1}^{[i]}$ 
37. end
38.  $J_k^{[i]} = (l + 1) J_{k|k-1}^{[i]}$ 
39. end

```

Table 2.2: Pruning procedure for PHD Filter

-
1. **Given:** $\{\eta_k^{[i,j]}, \mu_k^{[i,j]}, P_k^{[i,j]}\}_{j=1}^{J_k^{[i]}}$, a truncation threshold T , a merging threshold U
 2. and a maximum allowable Gaussian terms J_{max}
 3. set $l = 0$, and $I = \{i = 1 \dots J_k^{[i]} | \eta_k^{[i,j]} > T\}$
 4. **repeat**
 5. $q = \arg \max i \in I \eta_k^{[i,j]}$
 6. $L = \{i \in I | (\mu_k^{[i,j]} - \mu_k^{[i,q]})^T ((P_k^{[i,j]})^{-1}) (\mu_k^{[i,j]} - \mu_k^{[i,q]}) \leq U\}$
 7. $\hat{\eta}_k^{[i,l]} = \sum_{j \in L} \eta_k^{[i,j]}$
 8. $\hat{\mu}_k^{[i,l]} = (1/(\hat{\eta}_k^{[i,l]})) \sum_{i \in L} \eta_k^{[i,j]} \mu_k^{[i,j]}$
 9. $\hat{P}_k^{[i,l]} = (1/(\hat{\eta}_k^{[i,l]})) \sum_{i \in L} \eta_k^{[i,j]} (P_k^{[i,j]} + (\hat{\mu}_k^{[i,l]} - \mu_k^{[i,j]})(\hat{\mu}_k^{[i,l]} - \mu_k^{[i,j]})^T)$
 10. $I = I - L$
 11. **until** $I == \emptyset$
 12. if $l > J_{max}$ then replace $\{\hat{\eta}_k^{[i,l]}, \hat{\mu}_k^{[i,l]}, \hat{P}_k^{[i,l]}\}_{j=1}^l$ by those of the J_{max} Gaussians
 13. with largest weights.
 14. **Output:** $\{\hat{\eta}_k^{[i,l]}, \hat{\mu}_k^{[i,l]}, \hat{P}_k^{[i,l]}\}_{j=1}^l$ pruned Gaussian components.

Table 2.3: Particle filter based vehicle trajectory estimation

Sampling step:

- 1). For $i=1, \dots, N$, sample $\hat{X}_k^{[i]} \sim f_X(X_{k-1}^{[i]}, U_k)$ and set $\hat{w}_k^{[i]} = [(\lambda_c(z))^{\|Z_k\|} \exp(N_k - N_{k-1} - \lambda_c)] w_{k-1}^{[i]}$
- 2). Normalize weights.

Re-sampling step:

- 1). Resample $\{\hat{w}_k^{[i]}, \hat{X}_k^{[i]}\}_{i=1}^N$ to get $\{w_k^{[i]}, X_{0:k}^{[i]}\}_{i=1}^N$

it is clear that the Gaussians which corresponds to previously observed false features disappear from the PHD, while the magnitude of the Gaussian components increase as the resultant PHD is updated with new measurements. As it is clearly outlined in Table. 2.1, there is no explicit data association in the RFS SLAM solution, instead the PHD is updated with the measurements without measurements to landmark map association. The magnitude of the Gaussian components, which lie near the positions where observations are collected, increases while the magnitude of the Gaussians which corresponds to the false features reduces and being removed from the map if the magnitude is below a predetermined threshold. Those Gaussians, which have higher magnitudes (hence higher weights after the pruning process, see Table.2.2) are chosen as actual landmarks and their means are chosen as actual feature locations. This illustrates how the PHD filter estimate the landmark map avoiding explicit data association and filtering out measurement clutter, which are expensive sub-routines that are explicitly performed in the conventional SLAM solutions.

Particle filter based vehicle trajectory estimation In order to illustrate how particle filter works in RFS SLAM solution, let's consider the example where a single robot performs RFS Single Vehicle SLAM with four particles. Each particle corresponds to an estimate of its history of vehicle position, and there an associated map to each of these trajectories. Fig. 2.2 shows the maps and corresponding estimates of the vehicle trajectory of the four particles being used. The weight of each particle is shown below the corresponding map. Now, the map and the trajectory of the highest weighted particle is chosen as the best estimate of the vehicle trajectory and the corresponding map is chosen as the best estimate of the landmark map, in this example it is the second particle (Fig.2.2(b)).

(a) Map at time step $k-1$ (b) PHD at time step $k-1$ (c) Map at time step k (d) PHD at time step k (e) Map at time step $k+1$ (f) PHD at time step $k+1$ (g) Map at time step $k+2$ (h) PHD at time step $k+2$

Figure 2.1: How PHD of a landmark map evolves over time in a single vehicle RFS SLAM scenario. The figures on the left shows the map of the environment with the vehicle, where black stars are the actual landmarks and the black circles denote their estimations. The figures on the right shows the corresponding PHD at that time step.

Figure 2.2: In a RFS SLAM scenario, the landmark maps correspond to four particles are shown here. Actual vehicle trajectory of each particle is shown in red and the estimated trajectory is shown in blue. Actual landmark locations are shown as black stars, and their estimates are shown as black circles. The particle with the highest weight (particle 2) is chosen as the best estimate of the vehicle trajectory and the corresponding map is chosen as the best estimate of the landmark map.

2.3 Collaborative Multi-Vehicle SLAM algorithms

Researchers soon realized that deploying multiple autonomous robots, can greatly enhance the performance gains in terms of efficiency, accuracy and robustness of various applications such as search and rescue [1] [2] [3], port and harbor surveillance [4], hydrographic exploration [6], detection and tracking of moving objects [7] [8], intelligent transportation [9] [78] etc. Still, even with multi-vehicle systems, an essential requirement was to combine various sensory data acquired from multiple robot platforms with accurate positional information. Similar to the single vehicle case, in order to achieve improved performance gains, researchers discovered the necessity of developing CMSLAM algorithms.

2.3.1 Problem formulation

Let $U_{1:k}^{(r)} = [U_1^{(r)}, U_2^{(r)}, \dots, U_k^{(r)}]^T$ denote the time sequence of control commands applied to the vehicle r up to time k , where $U_k^{(r)}$ denotes the control command applied at time k . Let the time sequence of poses history of each vehicle be denoted by $X_{1:k}^{(r)} = [X_1^{(r)}, X_2^{(r)}, \dots, X_k^{(r)}]^T$, where $X_k^{(r)}$ denotes the pose of vehicle r with respect to the global reference frame, at time k . Let the time sequence of sets of measurements obtained using exteroceptive sensors mounted on each vehicle be denoted by $Z_{1:k}^{(r)} = [Z_1^{(r)}, Z_2^{(r)}, \dots, Z_k^{(r)}]$, where $Z_k^{(r)} = \{z_{k,1}^{(r)}, z_{k,2}^{(r)}, \dots, z_{k,n_k^{(r)}}^{(r)}\}$ denotes the measurement set received at time k , from vehicle r , while $n_k^{(r)}$ denotes the number of measurements. Let the map at time k be denoted by the set $M_k = \{m_{k,1}, m_{k,2}, \dots, m_{k,l_k}\}$, where l_k denotes the number of landmarks available in the map at time k .

Following the same vein as in the single vehicle case, there exist two major CMSLAM formulations, the online CMSLAM solution and the full CMSLAM solution. Let's consider the scenario where v number of robots performing CMSLAM.

The online CMSLAM algorithm

In general, all online CMSLAM solutions [79] [80] [81] stem from EKF-SLAM based solutions, where each collaborating robot solves the online mono-SLAM problem. Here, we are required to evaluate the posterior density,

$$p_{k|k}(X_k^{(1,\dots,v)}, M_k | Z_{1:k}^{(1,\dots,v)}, U_{1:k}^{(1,\dots,v)}, X_0^{(1,\dots,v)}) \quad (2.35)$$

where the abbreviations $Z_{1:k}^{(1,\dots,v)}$, $U_{1:k}^{(1,\dots,v)}$ and $X_{1:k}^{(1,\dots,v)}$ are given by,

$$Z_{1:k}^{(1,\dots,v)} = (Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, \dots, Z_{1:k}^{(v)}) \quad (2.36)$$

$$U_{1:k}^{(1,\dots,v)} = (U_{1:k}^{(1)}, U_{1:k}^{(2)}, \dots, U_{1:k}^{(v)}) \quad (2.37)$$

$$X_{1:k}^{(1,\dots,v)} = (X_{1:k}^{(1)}, X_{1:k}^{(2)}, \dots, X_{1:k}^{(v)}) \quad (2.38)$$

where current vehicle position estimate and the landmark map is propagated using a recursive Bayesian filtering framework.

The full CMSLAM algorithm

In general, all full CMSLAM solutions [82] [83] are based on the FastSLAM algorithm (particle filter based vehicle trajectory estimation). Here, we are required to evaluate the posterior density,

$$p_{k|k}(X_{1:k}^{(1,\dots,v)}, M_k | Z_{1:k}^{(1,\dots,v)}, U_{1:k}^{(1,\dots,v)}, X_0^{(1,\dots,v)}) \quad (2.39)$$

where the joint posterior consist of the history of the vehicle pose and the global landmark map is propagated using a recursive prediction and update procedure in order to solve the CMSLAM problem.

2.3.2 State of the art CMSLAM solutions

Fenwick et al. [79] presented one of the first algorithms on collaborative localization and mapping by extending the classical single vehicle EKF-SLAM implementation to multiple vehicles. Although this methodology could exploit the heterogeneity of sensing capabilities and incorporate vehicle to vehicle observations for improved accuracy, it is not scalable and suitable for mapping larger areas with larger number of autonomous vehicles, due to higher network bandwidth requirements and rapidly increasing computational resource requirements. A non-linear observability analysis of this solution can be found in [84].

Williams et al. proposed a scalable and efficient solution to the multi-robot SLAM problem [80] by extending their original work on constrained local sub-map filter (CLSF) [23] [24]. The key concept of CLSF is that the independent local sub-maps not only facilitated improved data association but also provided significant performance gains due to periodic map fusion. In the CLSF approach, instead of fusing every feature observation directly into the global map, a vehicle relies on the local sub-map from features observed in its vicinity. These independently developed sub-maps are merged in to the global map periodically by appropriately formulating constraints between two maps by identifying common features from overlapped areas. Although it doesn't address the problem of unknown initial correspondence, it provides a scalable efficient solution for the multi-vehicle SLAM problem.

Thrun et al. formulated a multi-robot SLAM algorithm based on the sparse extended information filters (SEIF-SLAM) [85] [22] which enables a team of robots to build a global map from local sub-maps of individual robots even if their relative starting locations are unknown and landmarks are ambiguous. SEIF constitutes an efficient solution to the SLAM problem, representing the SLAM posterior by a sparse Gaussian Markov random field (GMRF), which parametrizes the multivariate distribution through the information vector and the information matrix. Scalability was accomplished by alignment of the local maps into a single global map using a tree-based algorithm for searching "similar-looking" local landmark configurations, paired with a hill climbing algorithm.

Howard et al. [86] [87] developed a multi-robot SLAM algorithm based on manifold representation, which facilitates data association and loop closure detection with consistent maps. One notable design change in this approach is the delayed data association till it can be done with high confidence, making it more robust against false data association which is common in other SLAM algorithms. Relying

on centralized processing, makes it sensitive to communication failures, moreover extending into a large number robots is not feasible.

In [82] Howard et al. formulated a Rao-Blackwellised Particle Filter based approach (FastSLAM) to solve the multi-vehicle SLAM problem. During the rendezvous, maps from each robot are shared with one-another and fused by using the notion of virtual robots traveling backwards in time, in order to build a shared global map and estimate individual robot positions. This approach has the limitation of centralized processing, requiring a reliable communication link between the robots.

Fox [88] proposed a distributed multi-robot mapping and exploration approach relaxing the known initial positions requirement. Robots exchange relative position information, when they are in communication range of one another, and use this information to actively verify their location hypotheses using a rendezvous strategy, upon successful rendezvous, individual sub-maps are merged into a shared global map and used for estimation of individual robot locations.

Using EKF-SLAM as the basis, in [81] Zhou et al. presented a new solution for multi-robot SLAM problem, with relaxing known initial correspondence. During the rendezvous, mutual relative robot-to-robot range and bearing measurements are processed to compute the coordinate transformation between maps created independently by the robots in order to build a single global map, solving the map alignment problem.

Chang et al. [89] formulated a scalable solution to the multi-robot SLAM problem using topological/metric maps by decoupling the problem into smaller SLAM problems, using FastSLAM to build individual sub-maps. Minor limitation found in the proposed approach is the off-line optimization in topological maps, that leads to stop the exploration process temporarily before the robots resume.

In [90], Anderson et al. presented a collaborative smoothing and mapping algorithm for multi-robot systems based on the simultaneous smoothing and mapping (SAM) [91]. It is shown that the proposed solution is better equipped for non-linear process and measurement models than the conventional EKF-SLAM based solutions. It not only merge maps created by independent robots, but also recovers the complete vehicle trajectory of each robot.

In [92], Cunningham et al. presented a novel solution to the distributed multi-robot SLAM problem based on simultaneous smoothing and mapping (SAM) [91]. The proposed solution introduced a novel distributed data fusion (DDF) method for efficient, robust map distribution among collaborating robots to achieve results in terms of scalability, communication bandwidth requirements and node failures.

More recently, Carlone et al. [83] introduced a multi-robot SLAM algorithm based on the FastSLAM algorithm in limited communication, with relaxing initial correspondence. During the rendezvous, each robot exchange the proprioceptive and exteroceptive information recorded since the last rendezvous, which is used to perform reference frame transformations and update the individual estimations before restart performing grid based FastSLAM.

2.4 Conclusion

In this chapter we briefly discussed the state of the art Single-Vehicle SLAM solutions. Then, we have carefully examined the existing Mult-Vehicle SLAM solutions. By a closer inspection it can be seen that almost all the existing Multi-Vehicle SLAM solutions stem from conventional vector based SLAM solutions such as EKF-SLAM and FastSLAM. As a result, in addition to the actual SLAM filter, it is required to solve several additional sub-problems such as data association, clutter filtering

and landmark map management outside the Bayesian SLAM recursion in order to produce a consistent CMSLAM solution. Assumption of clutter free measurements and ideal landmark detection and data association limit the practical application of such solutions.

Most of the existing CMSLAM solutions are centralized solutions, as a result, it is impractical to extend such solutions into a large number of vehicles due to the exponentially increasing communication and computational bandwidth requirements. In addition, an autonomous Multi-Vehicle setup consists of large number of moving vehicles, requires precise detection and tracking of nearby moving vehicles as an essential requirement for safe navigation. None of the conventional SLAM solutions addresses this issue in an adequate manner and the focus is given only on the static landmarks (features) present in the sensor FOV. Moreover, although vehicle-vehicle observations are used to improve the accuracy of some of the existing Multi-Vehicle SLAM solutions, such solutions fail to provide additional information regarding the tracking process of the moving object in the sensor FOV, which we believe is an essential requirement for such solutions.

In the subsequent chapters of this thesis, we propose alternative solutions to the CMSLAM problem using the recently developed RFS SLAM filter framework [72] [73] in order address such issues and produce a practically applicable SLAM solution.

Chapter 3

Collaborative Multi-Vehicle SLAM using Random Finite Sets

3.1 Introduction

In this chapter we propose two new solutions to the Collaborative Multi-Vehicle SLAM (CMSLAM) problem using the RFS SLAM framework [72] [73]. Similar to FastSLAM [20] formulation, we start by factorizing the multi-vehicle SLAM posterior into a product of the joint vehicle trajectories posterior and the landmark map posterior conditioned on the history of vehicle poses. The landmark map and the measurements are modeled as RFSs [45], which permits dynamic estimation of landmark positions in the presence of measurement clutter, landmark detection uncertainty and data association uncertainty in a Bayesian filtering framework. The landmark map posterior is propagated using a Gaussian mixture (GM) implementation of a PHD filter [51] and the joint vehicle trajectories posterior is propagated using a Rao-Blackwellised particle filter [76].

Due to observation update from multiple sensors (from multiple robots), the PHD

update of the landmark map posterior is more complex than the RFS mono-SLAM solution [72] [73]. We adopt two recently developed multi-sensor information fusion approaches in multi-target tracking literature in order to address this problem in a statistically principled manner. The first method is the iterated PHD update [45] [48] method, and the second method is a computationally more expensive general multi-sensor fusion approach [93]. These two multi-sensor PHD update methods, combined with the particle filter [20] [76] based joint vehicle trajectories posterior estimation technique lies in the core of the proposed multi-vehicle SLAM solutions.

Section 3.2 formulates the multi-vehicle SLAM problem and present the system architecture of the proposed solution. In section 3.3, RFS formulation of measurement model and landmark map model is discussed. Section 3.4 explains the landmark map posterior propagation method. Implementation details and the joint path posterior estimation method is presented in Section 3.5. A computational complexity analysis is presented in Section 3.7. Extension to an arbitrary number of robots is discussed in Section 3.6. Results and analysis of the proposed approach is presented in Section 3.8, and finally Section 3.9 concludes the chapter.

3.2 Formulation of Probabilistic Random Finite Set Multi-Vehicle SLAM

Let $U_{1:k}^{(r)} = [U_1^{(r)}, U_2^{(r)}, \dots, U_k^{(r)}]^T$ denote the time sequence of control commands applied to the vehicle r up to time k , where $U_k^{(r)}$ denotes the control command applied at time k . Let the time sequence of poses history of each vehicle be denoted by $X_{1:k}^{(r)} = [X_1^{(r)}, X_2^{(r)}, \dots, X_k^{(r)}]^T$, where $X_k^{(r)}$ denotes the pose of vehicle r with respect to the global reference frame, at time k . Let the time sequence of sets of measurements obtained using exteroceptive sensors mounted on each vehicle be denoted by

$Z_{1:k}^{(r)} = [Z_1^{(r)}, Z_2^{(r)}, \dots, Z_k^{(r)}]$, where $Z_k^{(r)} = \{z_{k,1}^{(r)}, z_{k,2}^{(r)}, \dots, z_{k,n_k^{(r)}}^{(r)}\}$ denotes the measurement set received at time k , from vehicle r , while $n_k^{(r)}$ denotes the number of measurements. Let the map at time k be denoted by the set $M_k = \{m_{k,1}, m_{k,2}, \dots, m_{k,l_k}\}$, where l_k denotes the number of landmarks available in the map at time k . Without loss of generality, let's consider the case $r = 1, 2$ to make the exposition clearer.

In the Multi-Vehicle SLAM problem, we are required to evaluate the posterior probability distribution,

$$p_{k|k}(M_k, X_{1:k}^{(1)}, X_{1:k}^{(2)} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \quad (3.1)$$

for all times k , where $X_0^{(1)}$ and $X_0^{(2)}$ denote the initial poses of first and second vehicles respectively. In other words, it is required to evaluate the joint posterior density consisting of the map and the vehicles' pose history at all times k , conditioned on the time sequence of sets of observations, time sequence of control commands applied to individual vehicles up to and including time k and initial vehicle poses.

We make the following assumptions in the proposed CMSLAM solutions.

- Sufficient communication bandwidth is available at all times without delays.
- Initial vehicle positions are assumed to be known.

Initial vehicle position issue is in general solved by using the vehicle-vehicle observations. According to our perception, this is not an easy task to accomplish as it requires a precise moving target-tracking process run in parallel to the SLAM solution as the vehicles keep moving. None of the existing Multi-Vehicle SLAM solutions provides such a capability, but still assumes that the vehicle positions can be inferred using vehicle-vehicle observation. Although we do not address this issue explicitly, our work in Chapter 4 provides an adequate framework in order to solve

it in a practical application.

Multi-Vehicle SLAM system architecture

The three widely used system architectures for solving a typical multi-sensor information fusion problem are centralized, hierarchical and distributed [94]. Information fusion architectures in multi-vehicle SLAM follows the same vein.

In a centralized approach, all observations and control inputs acquired from the sensors mounted on each robot are communicated into a centralized node where the fusion takes place. In a typical hierarchical solution, each participating robot performs mono-SLAM and the resultant sub-maps are periodically fused by identifying the common features between the sub-maps and the existing global map, while aligning them to minimize the estimation errors. After fusion, each participating vehicle starts mono-SLAM by resetting their local maps. This approach is commonly used in EKF based mono-SLAM solutions due to the fact that they propagate the cross correlation between feature estimates. However a fully distributed architecture is problematic for solving the multi-vehicle SLAM problem due to the data incest problem.

In this work we use a centralized approach, emphasizing the statistically principled Multi-Vehicle data fusion for solving the multi-vehicle SLAM problem using RFSs.

RFS Multi-Vehicle SLAM filter

The posterior distribution (3.1) can be computed using a recursive prediction and update process. In other words, we evaluate the joint posterior density consisting of the map M_k and history of vehicle poses $X_{1:k}^{(r)}$, (for $r = 1, 2$) at time k , using existing map M_{k-1} , history of vehicle poses $X_{0:k-1}^{(r)}$ evaluated at time $k - 1$, and the time

sequence of sets of observations of each vehicle $Z_{1:k}^{(r)}$ and the control commands $U_{1:k}^{(r)}$ up to and including time k . Similar to Montemerlo's approach in FastSLAM [20], Multi-Vehicle SLAM posterior (3.1) can be factorized as a product of joint vehicle poses posterior and the map posterior conditioned on the vehicle poses as follows,

$$\begin{aligned} & p_{k|k}(M_k, X_{1:k}^{(1)}, X_{1:k}^{(2)} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \\ &= p_{k|k}(M_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\ &\times p_{k|k}(X_{1:k}^{(1)}, X_{1:k}^{(2)} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \end{aligned} \quad (3.2)$$

This decouples the Multi-Vehicle SLAM problem into two separate estimation problems. The key advantage of this factorization is two-fold, firstly it can benefit from using Monte Carlo methods (particle filtering) [20] [76] for joint vehicle trajectories estimation, making it possible to adopt complex non-linear motion models. Secondly and most importantly, it can benefit from using RFS representation and FISST techniques for map posterior estimation. By representing the map and measurements as RFSs and appropriately modeling map transition model, it is possible to dynamically estimate multiple landmarks in the presence of measurement clutter by taking data association, landmark detection and landmark survival uncertainties into account.

The proposed CMSLAM algorithm is shown as a block diagram in Fig. 3.1. Each collaborating vehicle produces observations which are communicated in to a centralized fusion node running a Gaussian Mixture implementation of a PHD filter (GM-PHD) [51]. These observations contain not only actual landmark generated measurements but also false measurements. The GM-PHD filter fuses these observations and extract a global map of the environment, which is then used to estimate the vehicle trajectories of the collaborating vehicles.

Figure 3.1: Proposed CMSLAM solution. A particle filter is used to estimate the joint vehicle trajectories. A GM-PHD filter is used to fuse the observations collected by individual collaborating robot and produce a single global map. The resultant map is then used to estimate the joint vehicle trajectories of the collaborating vehicles.

3.3 RFS formulation of measurement and landmark map models

Due to the motion of the two vehicles, at each time step, some of the landmarks that are already in the map will disappear in the limited field of view (FOV) of the exteroceptive sensors, while some others will continue to exist with new landmarks appearing in the sensor FOV, randomly changing the number of landmarks present in the map. The ordering of these landmarks brings no significance in solving the CMSLAM problem [73]. Hence the landmark map can be more naturally and effectively represented as a RFS, M_k ,

$$M_k = \{m_{k,1}, m_{k,2}, \dots, m_{k,l_k}\} \in \mathcal{F}(\mathcal{M}) \quad (3.3)$$

where each $m_k \in \mathcal{M}$ and $\mathcal{M}$ is the state space of the landmarks. $\mathcal{F}(\mathcal{M})$ is the collection of all finite subsets of $\mathcal{M}$.

Similarly, the ordering of measurements collected from the exteroceptive sensor of a vehicle carries no significance in solving the SLAM problem. Moreover, not all the measurements received are from actual landmarks, some of them are false alarms. Hence the measurements can be naturally represented as a RFS, $Z_k^{(r)}$,

$$Z_k^{(r)} = \{z_{k,1}^{(r)}, z_{k,2}^{(r)}, \dots, z_{k,n_k}^{(r)}\} \in \mathcal{F}(\mathcal{Z}^{(r)}) \quad (3.4)$$

where $\mathcal{F}(\mathcal{Z}^{(r)})$ is the collection of all finite subsets of the measurement space $\mathcal{Z}^{(r)}$ corresponds to r th vehicle's sensor measurements.

Landmark Map Transition Model

As the vehicles continue exploring the unknown environment, new observations are collected in the limited FOV of on-board sensors and fused into the map, causing new landmarks to appear in the map and some of the already existing landmarks to disappear. The new landmarks appearing in the joint sensor FOV of vehicles at time k are modeled as the RFS $\Gamma_k(X_k^{(1)}, X_k^{(2)})$. Let the RFS of map at time $k-1$ be denoted by M_{k-1} , then the map transition model at time k is given by,

$$M_k = \Gamma_k(X_k^{(1)}, X_k^{(2)}) \cup \left[\bigcup_{\zeta_{k-1} \in M_{k-1}} \Upsilon(\zeta_{k-1}) \right] \quad (3.5)$$

where, the Bernoulli RFS $\Upsilon(\zeta_{k-1})$ denotes the predicted state of the landmark $\zeta_{k-1} \in M_{k-1}$, which takes the form $\{\zeta_k\}$ with probability of reappearance $P_S(\zeta_{k-1})$ or $\emptyset$ with probability $1 - P_S(\zeta_{k-1})$. The probability density of state transition of the landmark from ζ_{k-1} to ζ_k conditioned on the probability of survival is given by

$f_M(\zeta_k | \zeta_{k-1}, X_k^{(1)}, X_k^{(2)})$. Note that, $\Gamma_k(X_k^{(1)}, X_k^{(2)})$ is independent of the landmarks that are reappearing from the map M_{k-1} . The corresponding RFS map transition density [73] can be written as,

$$f_M(M_k | M_{k-1}, X_k^{(1)}, X_k^{(2)}) = \sum_{W \subseteq M_k} f_M(W | M_{k-1}) f_\Gamma(M_k - W | X_k^{(1)}, X_k^{(2)}) \quad (3.6)$$

where, $f_M(W | M_{k-1})$ is the transition density of the set of features existing in the map, M_{k-1} from time $k-1$ to time k , while $f_\Gamma(M_k - W | X_k^{(1)}, X_k^{(2)})$ represents the density of the RFS $\Gamma_k(X_k^{(1)}, X_k^{(2)})$ of new landmarks appearing in the joint sensor FOV at time k .

Landmark Measurement Model

Measurements received from on-board exteroceptive sensors contain not only landmark generated measurements, but also clutter (false alarms) due to various factors such as transducer misalignments, manufacturing faults, adverse field conditions and so on. Moreover landmark detection uncertainties of sensors have to be taken into account in order to faithfully describe the landmark measurement model. By adopting RFS measurement representation, these limitations have been overcome by explicitly accounting for clutter, data association and landmark detection uncertainty within the SLAM filter framework.

Let M_k denote the landmark map, while $C_k^{(r)}$ denote the RFS representing clutter received from exteroceptive sensor mounted on the r th vehicle at time k , then corresponding measurements can be represented by the RFS,

$$Z_k^{(r)} = C_k^{(r)} \cup \left[\bigcup_{\zeta_k \in M_k} \Theta_k^{(r)}(\zeta_k) \right] \quad (3.7)$$

where $\Theta_k^{(r)}(\zeta_k)$ is a Bernoulli RFS representing the measurement corresponds to the landmark $\zeta_k \in M_k$. Due to the limited FOV of the sensor, $\Theta_k^{(r)}(\zeta_k)$ can have a value of the form $\{z_k^{(r)}\}$ with probability of detection $P_D^{(r)}(\zeta_k|X_k^{(r)})$ or $\emptyset$ with a probability of $1 - P_D^{(r)}(\zeta_k|X_k^{(r)})$. Note that the probability of detection, is a function of landmark state and vehicle position. Conditioned on detection, measurement likelihood function corresponds to the observation of landmark with state ζ_k by the r th vehicle is given by $g_k^{(r)}(z_k^{(r)}|\zeta_k, X_k^{(r)})$. The corresponding RFS measurement density can be written as,

$$g_k^{(r)}(Z_k^{(r)}|M_k, X_k^{(r)}) = \sum_{W \subseteq Z_k^{(r)}} g_Z^{(r)}(W|M_k, X_k^{(r)}) g_C^{(r)}(Z_k^{(r)} - W) \quad (3.8)$$

where, $g_Z^{(r)}(W|M_k, X_k^{(r)})$ denotes the density of the feature observations, while $g_C^{(r)}(Z_k^{(r)} - W)$ denotes the density of clutter.

Probability Density vs. Probability Hypothesis Density

In traditional estimation problems, a random vector based posterior density is propagated in time via its first and second order moments (mean and covariance). Analogously, instead of propagating the full posterior, a RFS based posterior density is approximately propagated in time via its intensity, the probability hypothesis density (PHD) [45].

Assume that the number of elements in a RFS M on $\mathcal{M}$ is Poisson distributed, with each of them is identically and independently distributed. Then the probability density of RFS M can be written as [45] [95],

$$p(M) = \frac{\prod_{\zeta \in M} v(\zeta)}{\exp(\int v(\zeta) d\zeta)} \quad (3.9)$$

where, v denotes the PHD of density $p(M)$. Total number of elements of M on a region $S \subseteq \mathcal{M}$ can be obtained as, $N = \int_S v(\zeta) d\zeta$ (for more information see [95]).

3.4 Landmark Map Posterior Estimation

Mahler [48] proposed the PHD filter as a computationally tractable approximation to the optimal multi-target Bayes filter, formulated using FISST techniques. In multi-target tracking, it is assumed that the sensor is static, while the targets are moving. On the other hand, in the SLAM problem, targets (landmarks) are assumed to be stationary, while the sensor is moving as it is mounted on the vehicle. As a result, PHD filter approximation can be adopted in SLAM in order to dynamically estimate the landmark locations.

In this section, we present the map posterior estimation method. The map posterior is first factorized into a recursive prediction/update steps and then propagated using a PHD filter. Two PHD filter update methods for solving the CMSLAM problem are also proposed based on two recently developed multi-sensor information fusion techniques in FISST.

In the case of Multi-Vehicle SLAM problem with two vehicles, the map posterior conditioned on the vehicle trajectories is given by,

$$p_{k|k}(M_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \quad (3.10)$$

and the corresponding PHD is given by,

$$D_{k|k}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) = \int p_{k|k}(M_k \cup \zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \delta M_k \quad (3.11)$$

where the integration with δ denotes a set integral. The PHD in (3.11) is essentially

described as the density of the expected number of landmarks in the map M_k , having the state ζ_k . The number of landmarks in the map M_k in a region S is given by,

$$N_{k|k} = \int_S D_{k|k}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) d\zeta_k \quad (3.12)$$

To propagate the PHD via recursive prediction and update steps, we factorize the map posterior (3.10) as follows,

$$\begin{aligned} p_{k|k}(M_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) &= g_k(Z_k^{(1)}, Z_k^{(2)} | M_k, X_k^{(1)}, X_k^{(2)}) \\ &\times \frac{p_{k|k-1}(M_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})}{l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})} \end{aligned} \quad (3.13)$$

where the denominator in (3.13) is the normalization constant and,

$$\begin{aligned} &p_{k|k-1}(M_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\ &= \int f_M(M_k | M_{k-1}, X_k^{(1)}, X_k^{(2)}) \\ &\times p_{k-1}(M_{k-1} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k-1}^{(1)}, X_{0:k-1}^{(2)}) \delta M_{k-1} \end{aligned} \quad (3.14)$$

Equations (3.14) and (3.13) are the map prediction and update posteriors respectively. Their PHDs can be recursively propagated using a PHD filter as follows.

3.4.1 PHD filter for map posterior estimation

PHD prediction step

The PHD of map prediction density (3.14) can be evaluated using the PHD of the map posterior density at time $k-1$ and the map transition model discussed in (3.6)

as follows,

$$\begin{aligned}
& D_{k|k-1}(\zeta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\
&= b_k(\zeta_k | X_k^{(1)}, X_k^{(2)}) \\
&+ \int P_S(\zeta_{k-1}) f_M(\zeta_k | \zeta_{k-1}, X_k^{(1)}, X_k^{(2)}) \\
&\times D_{k-1|k-1}(\zeta_{k-1} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k-1}^{(1)}, X_{0:k-1}^{(2)}) d\zeta_{k-1}
\end{aligned} \tag{3.15}$$

where, $b_k(\zeta_k | X_k^{(1)}, X_k^{(2)})$ denotes the intensity of new landmarks appearing in the joint sensor FOV. Total number of landmarks in the predicted map is given by,

$$N_{k|k-1} = \int D_{k|k-1}(\zeta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) d\zeta_k \tag{3.16}$$

If the environment is assumed to be static and the probability of survival of a landmark is constant, PHD of the map prediction (3.15) simplifies to,

$$\begin{aligned}
& D_{k|k-1}(\zeta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\
&= b_k(\zeta_k | X_k^{(1)}, X_k^{(2)}) \\
&+ P_S D_{k-1|k-1}(\zeta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k-1}^{(1)}, X_{0:k-1}^{(2)})
\end{aligned} \tag{3.17}$$

PHD update step

Assume that the number of false alarms (clutter) produced by the r th vehicle from a single scan over the sensor FOV is Poisson distributed with an average $\lambda_c^{(r)}$, and their physical distribution is given by $c_k^{(r)}(z_k^{(r)})$. Then, PHD of the map update posterior (3.13) can be obtained using either of the following two methods.

- Iterative update method

In this method, a two-fold approximation of the PHD filter update [45] is applied by iteratively updating the map posterior with each vehicle's observa-

tions. We start with factorizing the map update posterior (3.13) as follows,

$$\begin{aligned} p_{k|k}^{(1)}(M_k | Z_{1:k}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\ = \frac{g_k(Z_k^{(1)} | M_k, X_k^{(1)}) p_{k|k-1}(M_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})}{g_{k|k-1}(Z_k^{(1)} | Z_{1:k-1}^{(1)}, X_{0:k}^{(1)})} \end{aligned} \quad (3.18)$$

and

$$\begin{aligned} p_{k|k}^{(2)}(M_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\ = \frac{g_k(Z_k^{(2)} | M_k, X_k^{(2)}) p_{k|k}^{(1)}(M_k | Z_{1:k}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})}{g_{k|k-1}(Z_k^{(2)} | Z_{1:k-1}^{(2)}, X_{0:k}^{(2)})} \end{aligned} \quad (3.19)$$

where, in (3.18), the map posterior is first updated with the observations received from the first vehicle, and in (3.19), resultant posterior is updated with the observations received from the second vehicle. This approximation can be used as a computationally cheaper alternative to obtain the PHD of the map update posterior. For notational compactness we abbreviate the PHDs as follows.

$$D_{k|k-1}(\zeta_k) = D_{k|k-1}(\zeta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \quad (3.20)$$

$$D_{k|k}^{(1)}(\zeta_k) = D_{k|k}^{(1)}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \quad (3.21)$$

The detection probability of the sensor mounted on the r th vehicle is also abbreviated as,

$$P_D^{(r)} = P_D^{(r)}(\zeta_k | X_k^{(r)}) \quad (3.22)$$

Then, the PHD of the map update posterior is obtained iteratively as follows,

$$\begin{aligned} D_{k|k}^{(1)}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\ = (1 - P_D^{(1)}) D_{k|k-1}(\zeta_k) + \sum_{z_k^{(1)} \in Z_k^{(1)}} \frac{P_D^{(1)} l_{z_k^{(1)}}^{(1)}(\zeta_k) D_{k|k-1}(\zeta_k)}{1 + D_{k|k-1}[P_D^{(1)} l_{z_k^{(1)}}^{(1)}]} \end{aligned} \quad (3.23)$$

and

$$\begin{aligned} & D_{k|k}^{(2)}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)} X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\ &= (1 - P_D^{(2)}) D_{k|k}^{(1)}(\zeta_k) + \sum_{z_k^{(2)} \in Z_k^{(2)}} \frac{P_D^{(2)} l_{z_k^{(2)}}^{(2)}(\zeta_k) D_{k|k}^{(1)}(\zeta_k)}{1 + D_{k|k}^{(1)}[P_D^{(2)} l_{z_k^{(2)}}^{(2)}]} \end{aligned} \quad (3.24)$$

where for any function $h()$,

$$D_{k|k-1}[h()] = \int h(\zeta_k) D_{k|k-1}(\zeta_k) d\zeta_k \quad (3.25)$$

while $l_{z_k^{(1)}}^{(1)}$ and $l_{z_k^{(2)}}^{(2)}$ are given by,

$$l_{z_k^{(1)}}^{(1)}(\zeta_k) = \frac{g_k^{(1)}(z_k^{(1)} | \zeta_k, X_k^{(1)})}{\lambda_c^{(1)} c_k^{(1)}(z_k^{(1)})} \quad (3.26)$$

$$l_{z_k^{(2)}}^{(2)}(\zeta_k) = \frac{g_k^{(2)}(z_k^{(2)} | \zeta_k, X_k^{(2)})}{\lambda_c^{(2)} c_k^{(2)}(z_k^{(2)})} \quad (3.27)$$

The number of landmarks in the map after measurement update can be obtained by,

$$N_{k|k} = \int D_{k|k}^{(2)}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)} X_{0:k}^{(1)}, X_{0:k}^{(2)}) d\zeta_k \quad (3.28)$$

Although this approach is computationally efficient, it produces different PHDs depending on the ordering of sensor updates. In practice it is known that this doesn't produce significant differences when the probability of detection is high. But when the probability of detection is low, this can still produce significant differences in the resultant PHDs (see [96] for a complete analysis on this.)

- General multi-sensor update method

In this method, PHD of the map update posterior is obtained using the general multi-sensor PHD update method [93]. Given the PHD of the predicted map

posterior (3.15), corresponding PHD update can be obtained as follows,

$$D_{k|k}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) = (1 - P_D^{(1)})(1 - P_D^{(2)})D_{k|k-1}(\zeta_k) + \left[\sum_{\mathcal{P} \boxtimes_2 Z_k} \omega_{\mathcal{P}} \sum_{W \in \mathcal{P}} \rho_W(\zeta_k) \right] D_{k|k-1}(\zeta_k) \quad (3.29)$$

where the summation is taken over all so called "binary partitions" $\mathcal{P}$ of $Z_k = Z_k^{(1)} \cup Z_k^{(2)}$ (see [93] for more information). The notation, " $\mathcal{P} \boxtimes_2 Z_k$ " stands for " $\mathcal{P}$ partitions Z_k into binary cells W ", where $W \in \mathcal{P}$ has one of the following forms,

$$W = \{z_k^{(1)}\}, \quad W = \{z_k^{(2)}\}, \quad W = \{z_k^{(1)}, z_k^{(2)}\} \quad (3.30)$$

The values of $\rho_W(\zeta_k)$ and $\omega_{\mathcal{P}}$ are given by,

$$\rho_W(\zeta_k) = \begin{cases} \frac{P_D^{(1)} l_{z_k^{(1)}}^{(1)}(\zeta_k)(1 - P_D^{(2)})}{1 + D_{k|k-1}[P_D^{(1)} l_{z_k^{(1)}}^{(1)}(1 - P_D^{(2)})]} & \text{if } W = \{z_k^{(1)}\} \\ \frac{(1 - P_D^{(1)}) l_{z_k^{(2)}}^{(2)}(\zeta_k) P_D^{(2)}}{1 + D_{k|k-1}[(1 - P_D^{(1)}) l_{z_k^{(2)}}^{(2)} P_D^{(2)}]} & \text{if } W = \{z_k^{(2)}\} \\ \frac{P_D^{(1)} l_{z_k^{(1)}}^{(1)}(\zeta_k) P_D^{(2)} l_{z_k^{(2)}}^{(2)}(\zeta_k)}{D_{k|k-1}[P_D^{(1)} l_{z_k^{(1)}}^{(1)} P_D^{(2)} l_{z_k^{(2)}}^{(2)}]} & \text{if } W = \{z_k^{(1)}, z_k^{(2)}\} \end{cases} \quad (3.31)$$

and,

$$\omega_{\mathcal{P}} = \frac{\prod_{W \in \mathcal{P}} d_W}{\sum_{\mathcal{Q} \boxtimes_2 Z_k} \prod_{W \in \mathcal{Q}} d_W} \quad (3.32)$$

and,

$$d_W = \begin{cases} 1 + D_{k|k-1}[P_D^{(1)} l_{z_k^{(1)}}^{(1)}(1 - P_D^{(2)})] & \text{if } W = \{z_k^{(1)}\} \\ 1 + D_{k|k-1}[(1 - P_D^{(1)}) l_{z_k^{(2)}}^{(2)} P_D^{(2)}] & \text{if } W = \{z_k^{(2)}\} \\ D_{k|k-1}[P_D^{(1)} l_{z_k^{(1)}}^{(1)} P_D^{(2)} l_{z_k^{(2)}}^{(2)}] & \text{if } W = \{z_k^{(1)}, z_k^{(2)}\} \end{cases} \quad (3.33)$$

The resultant PHD (3.29) is invariant under different sensor reordering, but

this approach is computationally more demanding compared to the iterative update method. However, the computational load can be reduced by using sensor consistency gating [93], which prevents generation of partitions containing pairs of measurements that are unlikely to be generated from mutual feature observations. In addition we propose to improve the run-time performance by reducing the number of generated partitions as described in Section 3.7.

3.5 Implementation of the Multi-Vehicle SLAM algorithm

Montemerlo [20] proposed the first SLAM solution catering non-linear motion models, the FastSLAM algorithm, by propagating the vehicle trajectory posterior using a particle filter. In the same spirit, we propose propagating the joint vehicle trajectories posterior, using a particle filter.

In the case of Multi-Vehicle SLAM problem with two vehicles, the joint vehicle trajectories posterior at time k is given by,

$$p_{k|k}(X_{1:k}^{(1)}, X_{1:k}^{(2)} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \quad (3.34)$$

The density (3.34) is represented by a set of weighted particles denoted by Ω_k as follows,

$$\Omega_k = \left\{ w_k^{[i]}, X_{1:k}^{(1),[i]}, X_{1:k}^{(2),[i]} \right\}_{i=1}^{N_s} \quad (3.35)$$

where, $w_k^{[i]}$ is the weight of i th particle, while N_s denotes the number of particles. Similar to FastSLAM, the particle set is recursively propagated in time, using sequential importance resampling (SIR) approach as presented in Section 3.5.

The map is conditioned on the vehicle trajectories. Therefore, assuming that the vehicle trajectories are represented by a set of weighted particles (3.35) and the map is represented in its PHD form (3.11), the joint map and vehicle trajectories posterior (i.e. Multi-Vehicle SLAM posterior (3.1)) at time k can be represented by

$$\left\{ w_k^{[i]}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}, D_{k|k}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}) \right\}_{i=1}^{N_s} \quad (3.36)$$

where, N_s is the number of particles in the particle filter, while $w_k^{[i]}$ denotes the importance weight of i th particle. Note that each particle contains a PHD of the map, conditioned on its vehicle trajectories.

The vehicle trajectories are propagated using a particle filter, while the PHD of the map posterior is propagated using the Gaussian mixture implementation of the PHD filter (GM-PHD) [51].

3.5.1 Particle filter Implementation

Assume that the weighted set of particles Ω_{k-1} represent the joint vehicle trajectories posterior at time $k-1$. Then at time k , a new joint vehicle pose is sampled from each particle as follows,

$$(X_k^{(1),[i]}, X_k^{(2),[i]}) \sim f_X(X_k^{(1)}, X_k^{(2)} | X_{k-1}^{(1),[i]}, X_{k-1}^{(2),[i]}, U_k^{(1)}, U_k^{(2)}) \quad (3.37)$$

Note that, although (3.37) is given as a joint transition model, due to the conditional independence between two vehicles' motion, each vehicle pose can be sampled using their individual motion model as follows,

$$X_k^{(r),[i]} \sim f_X^{(r)}(X_k^{(r)} | X_{k-1}^{(r),[i]}, U_k^{(r)}) \quad (3.38)$$

The new vehicle poses, $(X_k^{(1),[i]}, X_k^{(2),[i]})$, are then added to the set of particles Ω_{k-1} , creating a temporary set of particles distributed according to the proposal distribution given by,

$$q_{k|k}(X_{1:k}^{(1),[i]}, X_{1:k}^{(2),[i]} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \quad (3.39)$$

Now, each particle in the temporary set is assigned an importance weight given by,

$$\begin{aligned} w_k^{[i]} &= \frac{\text{target distribution}}{\text{proposal distribution}} \\ &= \frac{p_{k|k}(X_{1:k}^{(1),[i]}, X_{1:k}^{(2),[i]} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)})}{q_{k|k}(X_{1:k}^{(1),[i]}, X_{1:k}^{(2),[i]} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)})} \end{aligned} \quad (3.40)$$

By factorizing the target distribution using Bayes rule and ignoring the normalization constant it can be shown that,

$$w_k^{[i]} \propto l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}) \quad (3.41)$$

where $l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})$ is the normalization constant in (3.13), which can be obtained by extending the approach proposed by Mullane et al. in their mono-SLAM solution [72] [73] by rearranging (3.13) as follows,

$$\begin{aligned} l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) &= g_k(Z_k^{(1)}, Z_k^{(2)} | M_k, X_k^{(1)}, X_k^{(2)}) \\ &\times \frac{p_{k|k-1}(M_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})}{p_{k|k}(M_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})} \end{aligned} \quad (3.42)$$

Note that (3.42) is true for all values of M_k . Therefore, without loss of generality, by assigning $M_k = \phi$,

$$\begin{aligned} l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) &= g_k(Z_k^{(1)}, Z_k^{(2)} | \phi, X_k^{(1)}, X_k^{(2)}) \\ &\times \frac{p_{k|k-1}(\phi | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})}{p_{k|k}(\phi | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})} \end{aligned} \quad (3.43)$$

Assuming the number of elements in M_k is Poisson distributed, we can re-write (3.43) using the definition (3.9) as follows,

$$\begin{aligned} l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) &= \prod_{z_k^{(1)} \in Z_k^{(1)}} \lambda_c^{(1)} c_k^{(1)}(z_k^{(1)}) \prod_{z_k^{(2)} \in Z_k^{(2)}} \lambda_c^{(2)} c_k^{(2)}(z_k^{(2)}) \\ &\times \exp(N_{k|k} - N_{k|k-1} - \lambda_c^{(1)} - \lambda_c^{(2)}) \end{aligned} \quad (3.44)$$

Note that, in this case all the measurements are considered as clutter (false alarms) since the map is empty. The clutter points are assumed to be independently and identically distributed in each vehicle's sensor FOV, with the number of points being Poisson distributed.

The importance weight of each particle in the temporary set is normalized such that, $\sum_{i=1}^{N_s} w_k^{[i]} = 1$. Then, a new set of N_s particles are drawn with replacement, where each particle is sampled with a probability proportional to its importance weight. The resultant particle set with their importance weights, denoted by Ω_k , represents the target posterior density (3.34).

In this thesis, stratified re-sampling [97] have been adopted on the particle set as the number of effective particles drop below a predefined threshold. However, re-sampling results in rapid loss of particle diversity (particle depletion), which can be reduced by replacing the re-sampling step with partial rejection control [98], where particles are drawn from several steps in the past rather than only from the

preceding time-step.

3.5.2 GM-PHD filter implementation

The PHD prediction and update equations, (3.17), (3.23) and (3.24) (iterative method), or (3.17) and (3.29) (general multi-sensor method) can be implemented using a GM-PHD filter [51]. Let the PHD of the map posterior at time $k - 1$, correspond to i th particle be denoted by a Gaussian mixture of the form,

$$D_{k-1|k-1}(\zeta_{k-1} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k-1}^{(1),[i]}, X_{0:k-1}^{(2),[i]}) = \sum_{j=1}^{J_{k-1}} \eta_{k-1}^{[i,j]} \mathcal{N}(\zeta_{k-1}; \mu_{k-1}^{[i,j]}, P_{k-1}^{[i,j]}) \quad (3.45)$$

where, J_{k-1} is the number of Gaussians, while $\eta_{k-1}^{[i,j]}$, $\mu_{k-1}^{[i,j]}$ and $P_{k-1}^{[i,j]}$ respectively denote weight, mean and covariance of the j^{th} Gaussian. Intensity of the new landmarks appearing in the joint sensor FOV of vehicles at time k can also be represented as a mixture of Gaussians as follows,

$$b_k(\zeta_k | X_k^{(1),[i]}, X_k^{(2),[i]}) = \sum_{j=1}^{J_{b,k}} \eta_{b,k}^{[i,j]} \mathcal{N}(\zeta_k; \mu_{b,k}^{[i,j]}, P_{b,k}^{[i,j]}) \quad (3.46)$$

where, $J_{b,k}$ denotes the expected number of new landmarks appearing in the joint FOV of the two vehicles. Therefore, PHD of the predicted map posterior (3.17) simplifies into a mixture of Gaussians of the form,

$$D_{k|k-1}(\zeta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}) = \sum_{j=1}^{J_{k|k-1}} \eta_{k|k-1}^{[i,j]} \mathcal{N}(\zeta_{k|k-1}; \mu_{k|k-1}^{[i,j]}, P_{k|k-1}^{[i,j]}) \quad (3.47)$$

Assume that the measurement model is given by a non-linear function of the form,

$$z_k^{(r)} = h_k^{(r)}(\zeta_k, X_k^{(r)}, \xi_k^{(r)}) \quad (3.48)$$

where $\xi_k^{(r)}$ is the zero-mean Gaussian noise with covariance $R_k^{(r)}$. We follow the extended Kalman filter (EKF) based measurement update method for the GM-PHD filter [51] by linearizing the measurement model and approximating the measurement likelihood function as follows,

$$g_k^{(r)}(z^{(r)}|\zeta_k, X_k^{(r)}) \approx \mathcal{N}(z^{(r)}; h_k^{(r)}(\zeta_k, X_k^{(r)}, 0), U_k R_k U_k^T) \quad (3.49)$$

where U_k is the Jacobian of $h_k^{(r)}(\zeta_k, X_k^{(r)}, \xi_k)$ with respect to ξ_k at $\xi_k = 0$. Let, $H_k^{(r)}$ denote the Jacobian of $h_k^{(r)}(\zeta_k, X_k^{(r)}, 0)$ with respect to ζ_k at some $\zeta_k = \mu$. Then we can show that the measurement updated PHD of the map posterior results in a mixture of Gaussians as follows.

Implementation of Iterative update method

Clearly, the first term in the PHD update equation (3.23) results in a mixture of Gaussians due to (3.47). By substituting (3.47) and (3.49), and using standard results of Gaussian functions [51], the second term in (3.23) can also be written as a mixture of Gaussians.

Hence, the PHD update equation due to the first vehicle's observations (3.23) results in a mixture of Gaussians as follows,

$$D_{k|k}^{(1)}(\zeta_k|Z_{1:k}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) = \sum_{j=1}^{J_{k|k}^{(1)}} \eta_{k|k}^{(1),[i,j]} \mathcal{N}(\zeta_{k|k}; \mu_{k|k}^{(1),[i,j]}, P_{k|k}^{(1),[i,j]}) \quad (3.50)$$

Starting with the resultant PHD (3.52), it can be shown that the PHD update equation due to the observations from the second vehicle (3.24) also results in a

mixture of Gaussians of the form,

$$D_{k|k}^{(2)}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) = \sum_{j=1}^{J_{k|k}^{(2)}} \eta_{k|k}^{(2),[i,j]} \mathcal{N}(\zeta_{k|k}; \mu_{k|k}^{(2),[i,j]}, P_{k|k}^{(2),[i,j]}) \quad (3.51)$$

where the values of $\eta_{k|k}^{(2),[i,j]}$, $\mu_{k|k}^{(2),[i,j]}$ and $P_{k|k}^{(2),[i,j]}$ can be obtained as outlined in Appendix A. The resultant PHD, abbreviated as $D_{k|k}^{(2)}(\zeta_k | \cdot)$, is an approximation to the actual PHD of the landmark map (3.11).

Implementation General multi-sensor update method

Again, the first term in the PHD update equation (3.29) produces a mixture of Gaussians due to (3.47). When W takes the value of the form of either $\{z_k^{(1)}\}$ or $\{z_k^{(2)}\}$, it can be shown that the second term also results in a mixture of Gaussians by following the same procedure as described in "Iterative update method". When W is of the form $W = \{z_k^{(1)}, z_k^{(2)}\}$, $\rho_W(\zeta_k)$ results in a constant, while the multiplication between $\rho_W(\zeta_k)$ and $D_{k|k-1}(\zeta_k)$ results in a mixture of Gaussians as follows,

$$\rho_W(\zeta_k) D_{k|k-1}(\zeta_k) = \sum_{j=1}^{J_{k|k-1}} \tilde{\eta}_{k|k}^{[i,j]} \mathcal{N}(\zeta_k; \mu_{k|k}^{(2),[i,j]}, P_{k|k}^{(2),[i,j]}) \quad (3.52)$$

where

$$\tilde{\eta}_{k|k}^{[i,j]} = \frac{\eta_{k|k-1}^{[i,j]} P_D^{(1)} P_D^{(2)} q_k^{(1),[i,j]}(z_k^{(1)}) q_k^{(2),[i,j]}(z_k^{(2)})}{\sum_{y=1}^{J_{k|k-1}} \eta_{k|k-1}^{[i,y]} P_D^{(1)} P_D^{(2)} q_k^{(1),[i,y]}(z_k^{(1)}) q_k^{(2),[i,y]}(z_k^{(2)})} \quad (3.53)$$

while $q_k^{(1),[i,j]}(z_k^{(1)})$ and $q_k^{(2),[i,j]}(z_k^{(2)})$ are abbreviations for $q_k^{(1),[i,j]}(z_k^{(1)}, X_k^{(1),[i]})$ and $q_k^{(2),[i,j]}(z_k^{(2)}, X_k^{(2),[i]})$ respectively. The values of $\mu_{k|k}^{(2),[i,j]}$, $P_{k|k}^{(2),[i,j]}$, $q_k^{(1),[i,j]}(z_k^{(1)})$ and $q_k^{(2),[i,j]}(z_k^{(2)})$ can be obtained as summarized in Appendix A. Hence it is apparent that the measurement updated PHD of the map posterior results in a mixture of

Gaussians of the following form.

$$D_{k|k}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) = \sum_{j=1}^{J_{k|k}} \eta_{k|k}^{[i,j]} \mathcal{N}(\zeta_k; \mu_{k|k}^{[i,j]}, P_{k|k}^{[i,j]}) \quad (3.54)$$

How GM-PHD filter works in CMSLAM

We take an example (Fig. 3.2) in order to illustrate how PHD filter works with the Multi-Vehicle scenario with mutual feature observations. Here, two vehicles performs RFS-CMSLAM-II from time step $k-1$ to $k+1$. At time $k-1$, the vehicles observe mutual features present in the joint sensor FOV and update the PHD and as a result, high magnitude peaks of the Gaussians can be observed in the joint sensor FOV due to mutual feature observations. The Gaussians located at (3,2.5), (3, -4), (5,-4) are due to the false feature observations. As we don't receive measurements from those features at time step k they are automatically removed from the map, since the weight of such Gaussians drop below a predefined threshold.

As it is clear, there is no explicit data association in the RFS CMSLAM solutions, instead the PHD is updated with the measurements without measurements to landmark map association. The magnitude of the Gaussian components, which lie near the positions where observations are collected, increases while the magnitude of the Gaussians which corresponds to the false features reduces and being removed from the map if the magnitude is below a predetermined threshold. Moreover, due to mutual observations, those landmarks which are at the joint sensor FOV, has much higher magnitudes than those landmarks which are not observed by both vehicles at the same time. Those Gaussians, which have higher magnitudes (hence higher weights after the pruning process) are chosen as actual landmarks and their means are chosen as actual feature locations. This illustrates how the PHD filter estimate the landmark map avoiding explicit data association and filtering out measurement

clutter, which are expensive sub-routines that are explicitly performed in the conventional SLAM solutions.

Figure 3.2: How PHD of the landmark map evolves over time, to recover the landmark map. In maps, black stars denote actual landmarks and the black circles denote their estimations. The yellow circles in the PHDs denote the actual landmark locations, and dotted lines denote the observations from each vehicle.

3.6 Multi-Vehicle SLAM with an arbitrary number of vehicles

The solution we have developed can be readily extended into an arbitrary number of robots. Consider the scenario where v number of robots performing multi-vehicle SLAM. The corresponding map posterior conditioned on the vehicle trajectories is given by,

$$p_{k|k}(M_k | Z_{1:k}^{(1,\dots,v)}, X_0^{(1,\dots,v)}) \quad (3.55)$$

and the joint vehicle trajectories posterior is given by,

$$p_{k|k}(X_{1:k}^{(1,\dots,v)} | Z_{1:k}^{(1,\dots,v)}, U_{1:k}^{(1,\dots,v)}, X_0^{(1,\dots,v)}) \quad (3.56)$$

where the abbreviations $Z_{1:k}^{(1,\dots,v)}$, $U_{1:k}^{(1,\dots,v)}$ and $X_{1:k}^{(1,\dots,v)}$ are given by,

$$Z_{1:k}^{(1,\dots,v)} = (Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, \dots, Z_{1:k}^{(v)}) \quad (3.57)$$

$$U_{1:k}^{(1,\dots,v)} = (U_{1:k}^{(1)}, U_{1:k}^{(2)}, \dots, U_{1:k}^{(v)}) \quad (3.58)$$

$$X_{1:k}^{(1,\dots,v)} = (X_{1:k}^{(1)}, X_{1:k}^{(2)}, \dots, X_{1:k}^{(v)}) \quad (3.59)$$

Under the same assumptions as presented in the two vehicle scenario, the map posterior in the v vehicle case can also be propagated using a PHD filter as follows.

3.6.1 PHD filter for map posterior estimation

PHD prediction step

The PHD of the predicted map posterior can be obtained using the PHD of the map posterior density at time $k - 1$, $D_{k-1|k-1}(\zeta_{k-1}|Z_{1:k-1}^{(1,\dots,v)}, X_{0:k-1}^{(1,\dots,v)})$ as follows,

$$\begin{aligned} D_{k|k-1}(\zeta_k|Z_{1:k-1}^{(1,\dots,v)}, X_{0:k}^{(1,\dots,v)}) &= b_k(\zeta_k|X_{0:k}^{(1,\dots,v)}) \\ &+ \int P_S(\zeta_{k-1})f_M(\zeta_k|\zeta_{k-1}, X_{0:k}^{(1,\dots,v)}) \\ &\times D_{k-1|k-1}(\zeta_{k-1}|Z_{1:k-1}^{(1,\dots,v)}, X_{0:k-1}^{(1,\dots,v)})d\zeta_{k-1} \end{aligned} \quad (3.60)$$

PHD update step

- Iterative update method

The PHD of the map posterior can be obtained by updating the PHD of the predicted map posterior (3.64) as follows,

$$\begin{aligned} D_{k|k}^{(1)}(\zeta_k|Z_{1:k}^{(1)}, Z_{1:k-1}^{(2,\dots,v)}, X_{0:k}^{(1,\dots,v)}) \\ = (1 - P_D^{(1)})D_{k|k-1}(\zeta_k) + \sum_{z_k^{(1)} \in Z_k^{(1)}} \frac{P_D^{(1)}l_{z_k^{(1)}}^{(1)}(\zeta_k)D_{k|k-1}(\zeta_k)}{1 + D_{k|k-1}[P_D^{(1)}l_{z_k^{(1)}}^{(1)}]} \end{aligned} \quad (3.61)$$

and

$$\begin{aligned} D_{k|k}^{(2)}(\zeta_k|Z_{1:k}^{(1,2)}, Z_{1:k-1}^{(3,\dots,v)}, X_{0:k}^{(1,\dots,v)}) \\ = (1 - P_D^{(2)})D_{k|k}^{(1)}(\zeta_k) + \sum_{z_k^{(2)} \in Z_k^{(2)}} \frac{P_D^{(2)}l_{z_k^{(2)}}^{(2)}(\zeta_k)D_{k|k}^{(1)}(\zeta_k)}{1 + D_{k|k}^{(1)}[P_D^{(2)}l_{z_k^{(2)}}^{(2)}]} \end{aligned} \quad (3.62)$$

by iteratively updating the PHD using measurements from all the vehicles we can obtain,

$$D_{k|k}^{(v)}(\zeta_k | Z_{1:k}^{(1,\dots,v)}, X_{0:k}^{(1,\dots,v)}) = (1 - P_D^{(v)})D_{k|k}^{(v-1)}(\zeta_k) + \sum_{z_k^{(v)} \in Z_k^{(v)}} \frac{P_D^{(v)} l_{z_k^{(v)}}^{(v)}(\zeta_k) D_{k|k}^{(v-1)}(\zeta_k)}{1 + D_{k|k}^{(v-1)}[P_D^{(v)} l_{z_k^{(v)}}^{(v)}]} \quad (3.63)$$

where the abbreviation $D_{k|k-1}(\zeta_k)$ represents the PHD of the predicted map posterior (3.64).

- General multi-sensor update method

Using the general multi-sensor PHD update method [93], PHD of the map update posterior can be obtained as follows,

$$D_{k|k}(\zeta_k | Z_{1:k}^{(1,\dots,v)}, X_{0:k}^{(1,\dots,v)}) = (1 - P_D^{(1)})(1 - P_D^{(2)}) \dots (1 - P_D^{(v)}) D_{k|k-1}(\zeta_k) + \left[\sum_{\mathcal{P} \in \mathcal{P}_v Z_k} \omega_{\mathcal{P}} \sum_{W \in \mathcal{P}} \rho_W(\zeta_k) \right] D_{k|k-1}(\zeta_k) \quad (3.64)$$

where unlike "binary partitions" in the two vehicle case, the summation is taken over all "v-ary" partitions $\mathcal{P}$ of $Z_k = Z_k^{(1)} \cup Z_k^{(2)} \cup \dots \cup Z_k^{(v)}$. In (3.68) $\rho_W(\zeta_k)$ and $\omega_{\mathcal{P}}$ are given as follows,

$$\rho_W(\zeta_k) = \begin{cases} \frac{P_{DL}^{([i_1])} Q_{DL}^{([i_1])}}{1 + D_{k|k-1}[P_{DL}^{([i_1])} Q_{DL}^{([i_1])}]} & \text{if } W = \{z_k^{(i_1)}\} \\ \frac{P_{DL}^{([i_1, i_2])} Q_{DL}^{([i_1, i_2])}}{1 + D_{k|k-1}[P_{DL}^{([i_1, i_2])} Q_{DL}^{([i_1, i_2])}]} & \text{if } W = \{z_k^{(i_1, i_2)}\} \\ \frac{P_{DL}^{([i_1, \dots, i_j])} Q_{DL}^{([i_1, \dots, i_j])}}{1 + D_{k|k-1}[P_{DL}^{([i_1, \dots, i_j])} Q_{DL}^{([i_1, \dots, i_j])}]} & \text{if } W = \{z_k^{(i_1, \dots, i_j)}\} \\ \frac{P_{DL}^{([1, \dots, v])}}{D_{k|k-1}[P_{DL}^{([1, \dots, v])}]} & \text{if } W = \{z_k^{(1, \dots, v)}\} \end{cases} \quad (3.65)$$

where,

$$P_{DL}^{([i_1, \dots, i_j])} = \prod_{u=i_1, u=i_2, \dots, u=i_j} P_D^{(u)} l_{z_k^{(u)}}^{(u)} \quad (3.66)$$

$$Q_{DL}^{([i_1, \dots, i_j])} = \prod_{u=1, u \neq i_1, \dots, i_j}^v (1 - P_D^{(u)}) \quad (3.67)$$

and,

$$\omega_{\mathcal{P}} = \frac{\prod_{W \in \mathcal{P}} d_W}{\sum_{\mathcal{Q} \in \mathcal{P}} \prod_{W \in \mathcal{Q}} d_W} \quad (3.68)$$

where,

$$l_{z_k^{(u)}}^{(u)}(\zeta_k) = \frac{g_k^{(u)}(z_k^{(u)} | \zeta_k, X_k^{(u)})}{\lambda_c^{(u)} c_k^{(u)}(z_k^{(u)})} \quad (3.69)$$

and,

$$\begin{aligned} d_W &= \begin{cases} 1 + D_{k|k-1}[P_{DL}^{(i_1)} Q_{DL}^{(i_1)}] & \text{if } W = \{z_k^{(i_1)}\} \\ 1 + D_{k|k-1}[P_{DL}^{(i_1, i_2)} Q_{DL}^{(i_1, i_2)}] & \text{if } W = \{z_k^{(i_1, i_2)}\} \\ 1 + D_{k|k-1}[P_{DL}^{(i_1, \dots, i_j)} Q_{DL}^{(i_1, \dots, i_j)}] & \text{if } W = \{z_k^{(i_1, \dots, i_j)}\} \\ D_{k|k-1}[P_{DL}^{(1, \dots, v)}] & \text{if } W = \{z_k^{(1, \dots, v)}\} \end{cases} \quad (3.70) \end{aligned}$$

where each i_j can take a value from $1, 2, \dots, v$. The length of the sequence $i_1, i_2, \dots, i_j$ is less than v with no two elements having the same value.

Similar to the two vehicle case, joint vehicle trajectories posterior in the v vehicle case can be propagated using a particle filter. The corresponding normalization constant can be obtained as,

$$\begin{aligned} & l_{k|k-1}(Z_k^{(1, \dots, v)} | Z_{1:k-1}^{(1, \dots, v)}, X_{0:k}^{(1, \dots, v)}) \\ &= \prod_{z_k^{(1)} \in Z_k^{(1)}} \lambda_c^{(1)} c_k^{(1)}(z_k^{(1)}) \dots \prod_{z_k^{(v)} \in Z_k^{(v)}} \lambda_c^{(v)} c_k^{(v)}(z_k^{(v)}) \\ & \times \exp(N_{k|k} - N_{k|k-1} - \lambda_c^{(1)} - \dots - \lambda_c^{(v)}) \end{aligned} \quad (3.71)$$

3.7 Computational Complexity

Let's consider the case where v number of collaborative robots performing multi-vehicle SLAM. Assume that the number of actual landmarks in the joint sensor FOV at time k is denoted by α_k and the total number of measurements received from sensors mounted on all collaborating vehicles is denoted by n_k . Then, the computational complexity of the CMSLAM solution based on the "Iterative update method" can be obtained as $O((v + \alpha_k n_k)N_s)$, where N_s denotes the number of particles used to propagate the v number of vehicle trajectories. In comparison, if the number of measurements received from each sensor is approximately equal to $\tilde{n}_k$, computational complexity of the CMSLAM solution based on the plain vanilla "General multi-sensor update method" is larger than $O((v + \alpha_k \tilde{n}_k!)N_s)$.

This heavy computational cost of the "General multi-sensor update method" based CMSLAM solution is mainly due to the combinatorial explosion of generated "v-ary" partitions. If the number of produced partitions is denoted by $|\mathcal{P}|$ the complexity can also be written as $O((v + \alpha_k |\mathcal{P}| n_k)N_s)$. In our solution we reduce this heavy cost by avoiding the creation of ineffective "v-ary" partitions.

In order to illustrate this, let's consider the case where two vehicles perform CMSLAM in the vicinity of each other and produce the observation sets $\{z_{k,1}^{(1)}, z_{k,2}^{(1)}\}$ and $\{z_{k,1}^{(2)}\}$ (Fig.3.3). In this case Z_k is given by,

$$Z_k = \left\{ z_{k,1}^{(1)}, z_{k,2}^{(1)} \right\} \cup \left\{ z_{k,1}^{(2)} \right\} \quad (3.72)$$

Figure 3.3: A mutual feature observation scenario. Blue triangles denote two robots while yellow stars denote features present in the environment. Both robots observe a common feature and produce corresponding observations $z_{k,2}^{(1)}$ and $z_{k,1}^{(2)}$.

and hence the corresponding binary partitions can be obtained as,

$$\begin{aligned}
 \mathcal{P}_1 &= \left\{ \{z_{k,1}^{(1)}\}, \{z_{k,2}^{(1)}\}, \{z_{k,1}^{(2)}\} \right\} \\
 \mathcal{P}_2 &= \left\{ \{z_{k,1}^{(1)}, z_{k,1}^{(2)}\}, \{z_{k,2}^{(1)}\} \right\} \\
 \mathcal{P}_3 &= \left\{ \{z_{k,2}^{(1)}, z_{k,1}^{(2)}\}, \{z_{k,1}^{(1)}\} \right\}
 \end{aligned} \tag{3.73}$$

Now let's say the measurements $z_{k,2}^{(1)}$ and $z_{k,1}^{(2)}$ are due to the observation of the same landmark by both vehicles, while $z_{k,1}^{(1)}$ is due to the observation of another distinct landmark. In this case we can safely exclude $\mathcal{P}_2$ from the set of binary partitions as it doesn't produce a significant contribution to the accuracy and proceed with partitions $\mathcal{P}_1$ and $\mathcal{P}_3$ in order to evaluate the resultant PHD using (3.29). Such mutual observations are identified by estimating the origins of the measurements using the current vehicle position estimates and setting a threshold on the distance among them. This method provides a significant performance gain as the number of measurements increases. Further performance improvements are possible by adopting the pruning and merging step for Gaussians as introduced in the GM-PHD filter [51].

Table 3.1: Parameters Used in the Simulation

Vehicle Parameters		Values
Velocity	V	2m/s
Sensor FOV	Range (r)	0 - 30m
	Bearing (b)	$-\pi$ - $+\pi$
Control Noise	Velocity (σ_v)	0.2m/s
	Steering Angle (σ_a)	2^0
Measurement Noise	Range (σ_r)	0.1m
	Bearing (σ_b)	1^0

3.8 Results and Analysis

3.8.1 Simulation Results

Simulations are conducted using the proposed CMSLAM algorithms (herein after referred to as RFS-CMSLAM-I and RFS-CMSLAM-II, respectively denoting the RFS-CMSLAM solution based on the iterative sensor update method and the general multi-sensor update method) in order to analyze the feasibility, utility and performance characteristics. The results are benchmarked against the CMSLAM algorithm (herein after referred to as RB-CMSLAM) proposed by Howard [82].

Two vehicles with identical control parameters are driven on two different trajectories on a square area of 200m x 200m with a set of 81 randomly placed landmarks. Each vehicle is equipped with a range-bearing sensor with identical field of view (FOV) and noise parameters. These parameters are summarized in Table 3.1.

Note that, control and sensor parameters are chosen such that they represent conditions in real applications. It is assumed that the number of false alarms produced by each sensor is Poisson distributed, and generated uniformly over the sensor FOV $[-\pi, \pi] \times [0, 30]$. Moreover, probability of detection of a landmark in the sensor FOV, $P_D^{(r)}(\zeta_k | X_k^{(r)})$ is assumed to be constant and set to .93, while the probability

Figure 3.4: Ground truth of the vehicle trajectories (vehicle 1 in blue and vehicle 2 in black) with randomly placed landmarks (black stars) superimposed on the set of accumulated false measurements (in green) recorded during a sample run under the clutter rate $\lambda_c^{(r)} = 5$, where $r = 1, 2$ denotes the vehicle number. Each vehicle starts at the origin, and closes the loop. The number of landmarks deployed in the simulation environment is 81.

of survival of a landmark in the map from time $k-1$ to k , $P_S(\zeta_{k-1}|X_k^{(1)}, X_k^{(2)})$ is also assumed to be constant and set to be .95. The control inputs (velocity and heading) are updated at 40Hz, while observations (range and bearing measurements) are updated at 5Hz. The simulation environment consisting of randomly placed landmarks and a set of collected measurement clutter from a sample run is shown in Fig.3.4 along with ground truth of vehicle trajectories.

Benchmarking RFS-CMSLAM against RB-CMSLAM

RB-CMSLAM algorithm is selected as a better candidate for performance comparison due to the close resemblance to the proposed RFS-CMSLAM algorithms. All three algorithms share the same standard particle filter based joint vehicle trajectories posterior propagation approach, but RB-CMSLAM algorithm is different from

the others in the representation of the map and the measurements, and the map propagation method. A total of 100 particles are used in all three algorithms for joint vehicle trajectory propagation.

In order to improve the performance under measurements with clutter, RB-CMSLAM algorithm is combined with a real-time implementation of feature elimination method using the negative evidence strategy proposed by Montemerlo [20]. In this simulation, log-odds ratio is incremented in steps of 0.01, and a threshold value of 0.05 is chosen for a feature to be confirmed as valid. These values are chosen in order to achieve comparable performance across varying clutter conditions.

Simulations are carried out with the clutter rate, $\lambda_c^{(r)} = 5$ and the performance of the proposed RFS-CMSLAM algorithms are benchmarked against the RB-CMSLAM algorithm. Fig.3.5-3.7 compare the estimated and the actual vehicle trajectories of a sample run of each algorithm. It can be seen that the RB-CMSLAM algorithm tends to produce inferior results producing considerably larger deviations in vehicle trajectory estimations from the ground truth, moreover a considerable amount of multiple detections (specially in overlapped vehicle trajectories) can also be seen. On the other hand, RFS-CMSLAM algorithms (RFS-CMSLAM-I and RFS-CMSLAM-II) produce superior results resulting in smaller deviations and consistent estimates of the features in both overlapped and non-overlapped vehicle trajectories. A comparison of vehicle position errors in terms of root mean squared error (RMSE) of 50 Monte Carlo (MC) runs is shown in Fig.3.8. As it can be seen, the performance of the RB-CMSLAM algorithm is much less satisfactory (loop closing error of more than 7m) compared to the proposed RFS-CMSLAM solutions (loop closing error of 3m). Moreover, it can also be seen that both RFS-CMSLAM algorithms produce almost identical results.

In Fig.3.9, actual running time (in seconds) of each algorithm at each time step

(a) RB-CMSLAM

Figure 3.5: Estimated vehicle trajectories (vehicle 1 in green and vehicle 2 in red) along with estimated feature locations (in cyan circles). A considerable amount of multiple feature detections (specially when the vehicle trajectories are overlapped) can be seen in the map. Moreover, estimated vehicle trajectories have large deviations from the ground truth (in blue and black).

(a) RFS-CMSLAM-I

Figure 3.6: Estimated vehicle trajectories (vehicle 1 in green and vehicle 2 in red) along with estimated feature locations (in cyan circles). Few multiple feature detections can be seen in the joint sensor FOV. Estimated vehicle trajectories have small deviations from the ground truth (in blue and black).

(a) RFS-CMSLAM-II

Figure 3.7: Estimated vehicle trajectories (vehicle 1 in green and vehicle 2 in red) along with estimated feature locations (in cyan circles). Much more consistent estimates of the landmarks can be seen everywhere in the map. It is also apparent that the estimated vehicle trajectories tends to produce much better results in the overlapped vehicle trajectories.

is compared. As it can be seen, both RFS-CMSLAM algorithms run faster than the RB-CMSLAM algorithm in most cases. In order to remove false features and improve the quality of the map, an adequately large log-odd threshold value is required. Hence performance of the RB-CMSLAM algorithm is largely influenced by the increasing clutter rate. On the other hand, RFS-CMSLAM algorithms' running times are largely unaffected except during the overlapping trajectories. In that case, RFS-CMSLAM-I algorithm outperforms the RFS-CMSLAM-II algorithm because of the larger number of mutual feature observations which results in a larger number of binary partitions. This in turn causes a larger number of Gaussian components to be propagated in the GM-PHD filter, making heavy computational demands.

Fig.3.10 compares the average number of landmarks available in the map. It is evident that the proposed RFS-CMSLAM algorithms produce much accurate estimations close to the actual number of observed landmarks, while benchmarked

(a) Vehicle 1

(b) Vehicle 2

Figure 3.8: RMS Vehicle position errors of 50 MC runs. Proposed RFS-CMSLAM solutions produce much smaller loop closing errors while the benchmarked RB-CMSLAM solution produces much larger errors under the same clutter conditions.

Figure 3.9: Running time of each algorithm at each simulation time step.

Figure 3.10: Average number of landmarks in the map, superimposed on the actual number of landmarks (in black). It can be seen that the benchmarked RB-CMSLAM algorithm produces a comparatively larger number of landmarks, while the proposed RFS-CMSLAM solutions produces much closer estimates to the number of actually observed landmarks.

RB-CMSLAM solution produces a larger estimation. Average map error (miss distance) over time is compared in Fig. 3.11 in terms of optimal sub-pattern assignment (OSPA) distance metric [77]. It can be seen that both RFS-CMSLAM algorithms produce smaller, map errors while RB-CMSLAM algorithm produces a larger map error. Moreover, due to the more consistent estimates in the overlapped areas, RFS-CMSLAM-II algorithm outperforms the RFS-CMSLAM-I algorithm in terms of map error.

Hence it is evident that the proposed RFS-CMSLAM algorithms produce superior results compared to the benchmarked RB-CMSLAM algorithm. It can also be seen that, although RFS-CMSLAM-I algorithm is theoretically suboptimal (due to the resultant variant PHDs under sensor reordering), simulation results are almost similar to the theoretically rigorous RFS-CMSLAM-II algorithm. Hence, RFS-CMSLAM-I algorithm can be chosen as the preferred choice when the robots are deployed in environments with overlapping vehicle trajectories with dense clutter conditions, due the heavy computational demands incurred in the RFS-CMSLAM-II algorithm. Having said that, RFS-CMSLAM-II algorithm produces much more consistent esti-

Figure 3.11: Average map error in terms of OSPA distance. Both RFS-CMSLAM solutions produce smaller map errors compared to the benchmarked RB-CMSLAM solution. It can also be seen that RFS-CMSLAM-II solution outperforms RFS-CMSLAM-I solution due to the much consistent feature estimation in the overlapped vehicle trajectories.

mates of the feature locations (specifically in the overlapped areas) and hence, tend to clearly outperforms RFS-CMSLAM-I algorithm in terms of map error.

Comparison with RFS-MONO-SLAM

For completeness, the performance of the proposed solutions are compared with RFS Single Vehicle SLAM [73] solution. A single vehicle is driven over each vehicle trajectory and the RMS vehicle position errors of 50MC runs are recorded and compared with the results produced by each vehicle during the RFS-CMSLAM solutions. To obtain comparable results, the number of particles used during the simulation of RFS-MONO-SLAM solution is 50. The results of the comparison is shown in Fig. 3.12. It is obvious that the proposed solutions Multi-Vehicle SLAM solution improve the accuracy of the vehicle trajectory estimation due make use of the overlapped areas which makes it possible to obtain advantage of mutual feature observations.

(a) Vehicle 1

(b) Vehicle 2

Figure 3.12: A comparison of RMS Vehicle position errors with RFS Single Vehicle SLAM (RFS-MONO-SLAM) of 50 MC runs. It can be seen that both proposed CMSLAM solutions produce slightly small RMS vehicle position errors compared to the single vehicle SLAM solution, which is mainly due to the overlapping areas of the simulation environment.

Figure 3.13: A picture of one of the robots used for the experiment. Each robot is equipped with a single axis gyroscope (KVH DSP 3000), a Laser scanner (SICK LMS 291) and wheel encoders. The control and measurement data are communicated into a centralized computer using a wireless network connection (Bullet2HP).

3.8.2 Experimental Results

Two identical mobile robots, each equipped with a Laser scanner, a single axis fiber optic gyroscope and a built-in wheel encoder (Fig.3.14) were used for the experiment. Both robots were simultaneously driven on a pre-planned vehicle paths (vehicle paths containing way-points at known locations) and the deviation of the estimated vehicle trajectories from these known way-points are used to evaluate the positional error. Each vehicle is driven over 400m on the respective vehicle path. The pillars present in the experimental area are used as landmarks.

The large pillar size prevents the detection of some other pillars present in the sensor FOV, as a result probability of detection of a landmark is set to .85, while the probability of survival is set to 0.90 in the proposed RFS-CMSLAM algorithms. The log-odd threshold for the RB-CMSLAM algorithm was set at 0.05. Mean velocity of the robots recorded during the experiment was approximately 0.4 m/s, which is

Figure 3.14: A picture of the robots taken during the experiment. The pillars present in the picture are used as features. Both vehicles are simultaneously driven on two vehicle trajectories and the control and observation data from each vehicle are communicated into a centralized computer.

erroneous and as a result the velocity standard deviation (std.), σ_v is set at 0.05 m/s, while steering angle std, σ_a is set at 2° . The observations are limited to the range from 0-20m and the bearing angle from -70° to 70° . Vehicle trajectories are estimated using 100 particles.

Fig.3.15 shows the estimated vehicle trajectories. As it can be seen, the proposed RFS-CMSLAM solutions produce much closer vehicle trajectory estimates to the ground truth, while the benchmarked RB-CMSLAM solution produces comparatively larger deviations. A comparison of the number of features present in the map is shown in Fig.3.16. It is clear that the proposed RFS-CMSLAM solutions produce closer estimates of the number of actual landmarks present in the environment, while RB-CMSLAM algorithm tends to produce a large number of false landmarks as well.

(a) RB-CMSLAM

(b) RFS-CMSLAM-I

(c) RFS-CMSLAM-II

Figure 3.15: A comparison of estimated vehicle trajectories (vehicle 1 in green and vehicle 2 in red), superimposed on a set of known way-points (vehicle 1 in green crosses and vehicle 2 in red crosses) on which the vehicles were driven. Estimated feature locations (in cyan circles) and approximate locations of actual features (in black stars) are also shown.

Figure 3.16: Average number of landmarks in the map. The number of pillars present in the experimental area was 36. It is clear that both RFS-CMSLAM solutions produce a closer estimate to the actual pillar count (number of actual landmarks), while RB-CMSLAM solution produces a larger estimate.

3.9 Conclusion

In this chapter we have presented a novel solution to the Collaborative Multi-Vehicle SLAM (CMSLAM) problem, using the recently developed SLAM filter framework based on random finite set (RFS) representation and finite set statistical (FISST) methods.

We propose to solve the CMSLAM problem by decomposing the the Multi-Vehicle SLAM posterior into the product of the landmark map posterior conditioned on the joint vehicle trajectories and the joint vehicle trajectories posterior. This decomposition allows propagation of the map posterior and the vehicle trajectories posterior as two different estimation problems. The joint vehicle trajectories posterior is propagated using a Rao-Blackwellized particle filter, where position of each vehicle is sampled using individual vehicle's motion model due to the conditional independence between their motion models. The landmark map posterior conditioned on the vehicle trajectories is recursively propagated by appropriately modeling the map and the measurements as RFSs and adopting FISST techniques. We have introduced two new multi-vehicle information fusion techniques for SLAM, yielding

two Bayesian CMSLAM algorithms with inbuilt map management, data association and clutter filtering, while catering for landmark detection uncertainty and data association uncertainty.

It is evident from the simulation results and experimental results that the proposed RFS-CMSLAM algorithms perform effectively under measurement clutter condition with data association uncertainty. Furthermore, Rao-Blackwellized particle filter based joint vehicle trajectories posterior estimation method enables adopting complex non-linear motion models, catering for various adverse field conditions that affect vehicular motion in challenging environments.

Chapter 4

Collaborative Multi-Vehicle SLAM with Moving Object Tracking

4.1 Introduction

Although up to date developments have made significant advancements towards practical application of SLAM solutions [13] [14] in autonomous mobile robot navigation, almost all existing formulations assume that the environment is static, ignoring the dynamic features present in the sensor field of view (FOV) as clutter. Although this assumption doesn't produce inferior results in the SLAM process at all times, it brings significant disadvantages into the autonomous navigation. For example if the moving features present in the sensor FOV are not being detected and tracked, such objects might be filtered out as clutter from the landmark map during the SLAM process. Hence any high-level application that makes use of the map for making navigation decisions could easily presume that it is safe to continue

the current course while dynamic objects are still present in the sensor FOV, leading to serious safety issues and limiting the practical applicability of SLAM algorithms. The most common solution for this issue is to consider the SLAM with detection and tracking of moving objects (DATMO) [38] [99] as two separate problems and solve them independent of each other at different levels. Although such a solution is adequate for almost all practical applications we believe that SLAM algorithms should be capable of solving both problems at the same time, providing a robust feature map (containing both static and dynamic feature information) for other high-level applications.

There have been few attempts in the robotics literature to overcome this issue and combine both SLAM and DATMO problems. The very first SLAM with DATMO solution was proposed by Wang et al. in [40], where both SLAM and DATMO are solved simultaneously and a practical application scenario is used to demonstrate the performance. In [39], Wang et al. extended their earlier work by presenting a theoretical framework for the SLAM with DATMO problem and demonstrated its performance using an EKF based implementation. Even though this approach can be used in some applications, it has a few drawbacks. Among them, decomposing the measurements into stationary objects and moving objects, inapplicability of complex multiple motion models per moving object are most notable. In order to classify measurements as stationary and dynamic, it is required to analyze the data over a sufficient period of time, using complex feature extraction algorithms. This may be undesirable in most practical applications if the results are to be used for making faster navigation decisions. In addition, most of the moving features such as pedestrians, animals or other vehicles may not have a single motion model, their motion models can be a mixture of motion models depending on the situation. Hence, the possibility of having multiple motion models per moving feature have to be taken into account. More recently Huijing et al. [42] also proposed a SLAM

solution combined with DATMO for dynamic large outdoor environments. The tracking process in this solution is not sensitive to changes in motion model of a moving object, which can result in false classification of a moving objects as a static objects frequently. Using Atlas framework [100] [101], Leedekerken et al. [102] describes an ongoing research work on combining real-time obstacle avoidance with SLAM for safe autonomous underwater navigation. A new obstacle avoidance technique is introduced based on the concept of virtual observations predicted from the resultant SLAM map.

As for an autonomous multi-vehicle setup, detection and tracking of moving objects can be considered as one of the most crucial requirement for practical applications. For example, in order to perform a hydrographic exploration using a set of autonomous surface crafts (ASCs) [10], each participating ASC must be capable of avoiding collision with each other and also with other moving surface crafts present in the environment. Moreover it is also required that the CMSLAM process prevents inclusion of dynamic features as static features in the map for improved the localization accuracy. In order to address these issues, in this Chapter, we propose a new algorithm, Collaborative Multi-Vehicle SLAM with detection and tracking of moving objects (herein after referred to as CMSLAMMOT), by extending our initial work on the RFS based CMSLAM solution (Chapter 3).

This chapter is organized as follows. Section 4.2 formulates the CMSLAMMOT problem and presents the solution architecture. Section 4.3 presents the RFS formulation of measurement and feature map models. Section 4.4 explains the static and dynamic feature map posteriors estimation method, while Section 4.5 presents the joint vehicle trajectories posterior estimation. Results and analysis of the proposed solution is presented in 4.6 using a simulation and an actual experiment. Section 4.7 concludes the chapter.

4.2 Formulation of Probabilistic Random Finite Set CMSLAM with Moving Object Tracking

Let $U_{1:k}^{(r)} = [U_1^{(r)}, U_2^{(r)}, \dots, U_k^{(r)}]^T$ denote the time sequence of control commands applied to the vehicle r up to time k , where $U_k^{(r)}$ denotes the control command applied at time k . Let the time sequence of pose history of each vehicle be denoted by $X_{1:k}^{(r)} = [X_1^{(r)}, X_2^{(r)}, \dots, X_k^{(r)}]^T$, where $X_k^{(r)}$ denotes the pose of vehicle r with respect to the global frame of reference, at time k . Let the time sequence of sets of measurements obtained using exteroceptive sensors mounted on each vehicle be denoted by $Z_{1:k}^{(r)} = [Z_1^{(r)}, Z_2^{(r)}, \dots, Z_k^{(r)}]$, where $Z_k^{(r)} = \{z_{k,1}^{(r)}, z_{k,2}^{(r)}, \dots, z_{k,n_k^{(r)}}^{(r)}\}$ denotes the measurement set received at time k , from vehicle r , while $n_k^{(r)}$ denotes the number of measurements. Let the augmented feature map at time k be denoted by the set $M_k = \{m_{k,1}, m_{k,2}, \dots, m_{k,l_k}\}$, where l_k denotes the number of features available in the map at time k and each m_k denotes the state of a feature. Without loss of generality, let's consider the case $r = 1, 2$ to make the exposition clearer.

Unlike earlier approaches to solve the SLAM problem, the CMSLAMMOT problem requires a broader definition for the state of a feature. In conventional SLAM solutions, the terms, feature and landmark are interchangeably used in order to specify a static distinguishable object whose position with respect to the vehicle can be extracted from the sensor measurements. But in CMSLAMMOT, we adopt the term feature in order to specify a distinguishable object present in the environment, which can either be a static feature or a moving (dynamic) feature. In addition, these dynamic features could also change their motion model. For example consider the motion of a pedestrian; a pedestrian could walk straight, stop, change the walking direction and might also change the speed and so on. Hence, the proposed CMSLAMMOT solution is expected to address such issues and provide a robust so-

lution for detection and tracking of moving objects present in the joint sensor FOV in addition to performing CMSLAM.

In the CMSLAMMOT problem, we are required to evaluate the two posterior probability distributions,

$$p_{k|k}(S_k, X_{1:k}^{(1)}, X_{1:k}^{(2)} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \quad (4.1)$$

and,

$$p_{k|k}(D_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \quad (4.2)$$

where, the RFS S_k denotes the set of static features present in the feature map at time k and the RFS D_k denotes the set of dynamic features present in the current sensor FOV. The distribution (4.1) is the familiar CMSLAM posterior and the distribution (4.2) is the dynamic feature map posterior conditioned on the history of vehicle poses. In our solution, we track both static and dynamic features present in the current sensor FOV in the presence of measurement clutter, landmark detection uncertainty and data association uncertainty and propose a Bayesian solution in order to evaluate (4.1) and (4.2) simultaneously.

Let the RFS M_k denote the augmented feature map, which consists of both static features observed by vehicles during their navigation and the dynamic features present in the current joint sensor FOV. Hence we can re-write M_k as $M_k = S_k \uplus D_k$ and the corresponding augmented feature map posterior conditioned on the history of vehicle poses can be written as,

$$p_{k|k}(M_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) = p_{k|k}(S_k \uplus D_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \quad (4.3)$$

Dynamic behavior of a feature is dependent on its type, for example a static feature might always remain at the same location while a dynamic feature might change

its location depending on the type of motion model being used. Moreover, some of the features might change the type of motion model, for example a dynamic feature could become a static feature for a short period of time and then switch back to another motion model and so on (Eg. pedestrians, vehicles etc.). Hence in order to faithfully perform CMSLAMMOT, we model the dynamics of the augmented feature map posterior (4.3) and cater for features having complex dynamic behaviors (with switching multiple motion models).

We make the same assumptions as discussed in the proposed CMSLAM solution (Chapter 3) in order to formulate the CMSLAMMOT solution. For completeness we summarize the assumptions as follows.

- Sufficient communication bandwidth is available at all times without delays.
- Initial vehicle positions are assumed to be known.

Although we do not solve the initial vehicle position issue, the theory we present in this chapter can be used to come up with a practical CMSLAM solution with unknown initial vehicle positions. In order to solve the unknown initial vehicle position issue, it requires precise moving target-tracking. Each collaborating vehicle is required to employ a precise multi-target tracker which tracks specific motion models (uniquely identify collaborating vehicles from other moving objects) and use that information in order to derive the vehicle positions. In the proposed CMSLAMMOT solution, we propose detection and tracking of moving objects having multiple complex motion models, present in the joint sensor FOV. Hence, it is possible to estimate the positions of the collaborating vehicles in the vicinity and derive their locations with respect to the global frame of reference solving the initial vehicle position issue.

CMSLAMMOT system architecture

We follow a centralized approach similar to the proposed RFS CMSLAM solution (Chapter 3), emphasizing the statistically principled data fusion for solving the CMSLAMMOT problem using RFSs. All observations and control inputs acquired from the sensors mounted on each robot are communicated into a centralized fusion node, where the fusion is taken place. Extracted static and dynamic feature maps along with the estimated vehicle trajectories can be communicated back into each collaborating robot for making navigation decisions at various levels.

RFS CMSLAMMOT filter

We solve the CMSLAMMOT problem using a recursive prediction/update procedure similar to the approach proposed in the RFS CMSLAM solution (Chapter 3). The joint posterior density consisting of the static feature map S_k and the history of vehicle poses $X_{1:k}^{(r)}$ is propagated using the existing static feature map S_{k-1} , history of vehicle poses $X_{0:k-1}^{(r)}$ evaluated at time $k-1$, the time sequence of sets of observations of each vehicle $Z_{1:k}^{(r)}$ and the applied control commands $U_{1:k}^{(r)}$ up to and including time k . However unlike the RFS CMSLAM solution, the transition model of the static feature map posterior is not uni-model, meaning, the static features present in the current sensor FOV can become dynamic features, while the dynamic features present in the current sensor FOV can become static features in the following time step.

Let's start with Montemerlo's approach in FastSLAM [20] to factorize the posterior (4.1) into a product of the history of vehicle poses posterior and the static feature

map posterior conditioned on the history of vehicle poses as follows,

$$\begin{aligned}
 & p_{k|k}(S_k, X_{1:k}^{(1)}, X_{1:k}^{(2)} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \\
 &= p_{k|k}(S_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{1:k}^{(1)}, X_{1:k}^{(2)}) \\
 &\times p_{k|k}(X_{1:k}^{(1)}, X_{1:k}^{(2)} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)})
 \end{aligned} \tag{4.4}$$

The joint vehicle trajectories posterior can be easily propagated using a particle filter provided that the static feature map posterior is available. But unlike the CMSLAM solution, we do not propagate a separate static feature map posterior and instead we propagate an augmented feature map posterior (4.3) which allows dynamic estimation of features having a combination of multiple motion models. Hence, in order to obtain the static feature map posterior, we proceed by marginalizing the augmented feature map posterior as (4.3) as follows,

$$p_{k|k}(S_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) = \int p_{k|k}(S_k \uplus D_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \delta D \tag{4.5}$$

It is clear that the static feature map posterior (4.5) is computationally intractable as it is - hence an alternative approach is required in order to obtain the static feature map posterior.

In order to propagate the augmented feature map posterior (4.3), we employ a PHD filter accommodating for dynamic estimation of features, which can switch the motion models. Hence the resultant PHD is a combination of PHD components, consisting of a PHD for each feature type. As a result now it is possible to extract the PHDs that correspond to dynamic feature map posterior (4.2) and the static feature map posterior (4.5). Hence it is possible to simultaneously estimate the dynamic feature locations and solve the CMSLAM problem.

The proposed CMSLAMMOT algorithm is shown as a block diagram in Fig. 4.1.

Figure 4.1: Proposed CMSLAMMOT solution. A particle filter is used to estimate the joint vehicle trajectories. A GM-PHD filter capable of handling switching multi-model motion is used to fuse the observations collected by individual collaborating robot and produce both a static feature map map and a dynamic feature map. The resultant static feature map is then used to estimate the joint vehicle trajectories of the collaborating vehicles, and the dynamic feature map is used in the navigation stack to avoid collisions with dynamic objects present in the sensor FOV.

Each collaborating vehicle produces observations which are communicated into a centralized fusion node running a Gaussian Mixture implementation of a PHD filter (GM-PHD) [51], which is capable of handling switching between different motion models. The observations contain measurements from both static and dynamic features present in the vicinity. Moreover, clutter can also be present in the measurement set due to various reasons. The proposed GM-PHD filter, tracks both static and dynamic features and caters switching between different motion models as well. The resultant static feature map is used to estimate the vehicle trajectories, while the dynamic feature map is used to avoid collision with dynamic objects present in the sensor FOV.

4.3 RFS formulation of measurement and feature map models

The measurements acquired during the CMSLAMMOT process contains both actual measurements (static and dynamic) and the false measurements. The ordering of these measurements carries no significance in solving the CMSLAMMOT problem. Hence the measurements can be naturally represented as a RFS, $Z_k^{(r)}$,

$$Z_k^{(r)} = \{z_{k,1}^{(r)}, z_{k,2}^{(r)}, \dots, z_{k,n_k}^{(r)}\} \in \mathcal{F}(\mathcal{Z}^{(r)}) \quad (4.6)$$

where $\mathcal{F}(\mathcal{Z}^{(r)})$ is the collection of all finite subsets of the measurement space $\mathcal{Z}^{(r)}$ corresponds to r th vehicle's sensor measurements.

Unlike the proposed RFS CMSLAM solution, we propagate an augmented RFS, M_k consisting of both static features observed during the SLAM process and the dynamic features present in the current joint sensor FOV and estimate both static and dynamic features in order to solve the CMSLAMMOT problem. The augmented feature map can be more naturally and effectively be represented as the RFS, M_K ,

$$M_k = \{m_{k,1}, m_{k,2}, \dots, m_{k,l_k}\} \in \mathcal{F}(\mathcal{M}) \quad (4.7)$$

where each $m_k \in \mathcal{M}$ and $\mathcal{M}$ is the state space of the features. $\mathcal{F}(\mathcal{M})$ is the collection all finite subsets of $\mathcal{M}$. In order to cater for features with multiple motion models, we extend the state of a feature from the conventional SLAM solution to a broader definition as follows.

Let $m_{k,i}$ be the state of any feature in the feature map at time k , then $m_{k,i}$ is given by $(\zeta_{k,i}, \beta_{k,i})$ where $\zeta_{k,i}$ is the kinematic state and $\beta_{k,i}$ is the type of the feature. Features are classified based on their motion model. For example, static features

are classified as type 0, i.e. $\beta_k = 0$, while moving features with motion model I, are of type 1, i.e. $\beta_k = 1$ and moving features with motion model II are of type 2, i.e. $\beta_k = 2$ and so on. We can consider that the motion of each moving object can be decomposed into a set of known motion models at various time intervals. In addition, features are allowed to switch between different motion models (types) over time allowing a comprehensive treatment for dynamic objects. This behavior is modeled using a jump Markov transition model, catering for changes in motion models with varying probabilities.

Augmented RFS Feature Map Transition Model

Collaborating vehicles collect observations of both static and dynamic features present in the environment, and communicated into a centralized fusion node, where the augmented feature map is recovered. Unlike our CMSLAM solution where we accommodated for only static features, in CMSLAMMOT, both static and dynamic features are considered. In addition, each feature type is allowed to switch their dynamics between different motion models as well.

The newly appearing features in the joint sensor FOV at time k are modeled as the RFS $\Gamma_k(X_k^{(1)}, X_k^{(2)})$. Let the RFS of landmark map at time $k - 1$ be denoted by M_{k-1} , then the landmark map transition model at time k can be written as,

$$M_k = \Gamma_k(X_k^{(1)}, X_k^{(2)}) \cup \left[\bigcup_{\xi_{k-1} \in M_{k-1}} \Upsilon(\xi_{k-1}) \right] \quad (4.8)$$

where, the Bernoulli RFS $\Upsilon(\xi_{k-1})$ denotes the predicted state of the landmark $\xi_{k-1} \in M_{k-1}$, which takes the form $\{(\zeta_k, \beta_k)\}$ with probability of reappearance $P_S(\zeta_{k-1}, \beta_{k-1})$ or $\emptyset$ with probability $1 - P_S(\zeta_{k-1}, \beta_{k-1})$. The probability density of state transition from $(\zeta_{k-1}, \beta_{k-1})$ to (ζ_k, β_k) , conditioned on the probability of

survival is given by, $f_M(\zeta_k, \beta_k | \zeta_{k-1}, \beta_{k-1}, X_k^{(1)}, X_k^{(2)})$. Note that, $\Gamma_k(X_k^{(1)}, X_k^{(2)})$ is independent of the landmarks that are reappearing from the map M_{k-1} . The corresponding RFS map transition density can be written as,

$$f_M(M_k | M_{k-1}, X_k^{(1)}, X_k^{(2)}) = \sum_{W \subseteq M_k} f_M(W | M_{k-1}) f_\Gamma(M_k - W | X_k^{(1)}, X_k^{(2)}) \quad (4.9)$$

where, $f_M(W | M_{k-1})$ is the transition density of the set of features existing in the map, M_{k-1} from time $k-1$ to time k , while $f_\Gamma(M_k - W | X_k^{(1)}, X_k^{(2)})$ represents the density of the RFS $\Gamma_k(X_k^{(1)}, X_k^{(2)})$ of newly appearing landmarks in the joint sensor FOV at time k .

In order to track features with multiple motion models, we adopt the same method as presented in [103] and decompose the transition density as follows.

$$\begin{aligned} f_M(\zeta_k, \beta_k | \zeta_{k-1}, \beta_{k-1}, X_k^{(1)}, X_k^{(2)}) \\ = \hat{f}_M(\zeta_k | \zeta_{k-1}, \beta_{k-1}, X_k^{(1)}, X_k^{(2)}) t_{k|k-1}(\beta_k | \beta_{k-1}) \end{aligned} \quad (4.10)$$

where $t_{k|k-1}(\beta_k | \beta_{k-1})$ denotes the Markov transition probability of switching mode from β_{k-1} to β_k .

RFS Measurement Model

Measurements received from exteroceptive sensors mounted on each vehicle are fused into the augmented feature map posterior, which consist of static features observed during the SLAM process and the dynamic features present in the current joint sensor FOV. New features could enter the augmented feature map and coexist with the already existing features while some of the features already in the map (and in the current FOV) could disappear from the map.

Let M_k denote the augmented feature map, while $C_k^{(r)}$ denote the RFS representing clutter received from exteroceptive sensor mounted on r th vehicle at time k , then corresponding measurements can be represented by the RFS,

$$Z_k^{(r)} = C_k^{(r)} \cup \left[\bigcup_{\xi_k \in M_k} \Theta_k^{(r)}(\xi_k) \right] \quad (4.11)$$

where $\Theta_k^{(r)}(\xi_k)$ is a Bernoulli RFS representing the measurement corresponding to feature $\xi_k \in M_k$ having state (ζ_k, β_k) . Due to the limited FOV of the sensor, $\Theta_k^{(r)}(\xi_k)$ can have a value of the form $\{z_k^{(r)}\}$ with probability of detection $P_D^{(r)}(\zeta_k, \beta_k | X_k^{(r)})$ or $\emptyset$ with a probability of $1 - P_D^{(r)}(\zeta_k, \beta_k | X_k^{(r)})$. Note that the probability of detection is a function of landmark state and vehicle position. Conditioned on detection, measurement likelihood function corresponds to the observation of feature with state (ζ_k, β_k) by the r th vehicle is given by $g_k^{(r)}(z^{(r)} | \zeta_k, \beta_k, X_k^{(r)})$. The corresponding RFS measurement density can be written as,

$$g_k^{(r)}(Z_k^{(r)} | M_k, X_k^{(r)}) = \sum_{W \subseteq Z_k^{(r)}} g_Z^{(r)}(W | M_k, X_k^{(r)}) g_C^{(r)}(Z_k^{(r)} - W) \quad (4.12)$$

where, $g_Z^{(r)}(W | M_k, X_k^{(r)})$ denotes the density of the feature observations, while $g_C^{(r)}(Z_k^{(r)} - W)$ denotes the density of clutter.

4.4 Estimation of Static and Dynamic Feature Map Posteriors

In the CMSLAMMOT problem, we are required to detect and track both static and dynamic features in the presence of measurement clutter, feature detection uncertainty and data association uncertainty, while accommodating switching of

motion models. In this section, we propose to propagate the augmented feature map posterior using a PHD filter and estimate the static and dynamic feature map posteriors using the resultant PHD.

Let's start with the augmented feature map posterior conditioned on the history of vehicle poses, which given by,

$$p_{k|k}(M_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \quad (4.13)$$

and the corresponding PHD is given by,

$$D_{k|k}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) = \int p_{k|k}(M_k \cup \zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \delta M_k \quad (4.14)$$

where the integration with δ denotes a set integral. The PHD (1.14) is the density of the expected number of features (static and dynamic) present in the augmented feature map M_k , having the state ζ_k . The number of total features present in the augmented feature map in a region S is given by,

$$N_{k|k} = \int_S D_{k|k}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) d\zeta_k \quad (4.15)$$

In order to propagate the augmented feature map posterior using a PHD filter [45] [48], we proceed with factorizing the posterior (4.13) as follows,

$$\begin{aligned} p_{k|k}(M_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) &= g_k(Z_k^{(1)}, Z_k^{(2)} | M_k, X_k^{(1)}, X_k^{(2)}) \\ &\times \frac{p_{k|k-1}(M_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})}{l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})} \end{aligned} \quad (4.16)$$

where the denominator in (4.16) is the normalization constant and,

$$\begin{aligned}
 & p_{k|k-1}(M_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\
 &= \int f_M(M_k | M_{k-1}, X_k^{(1)}, X_k^{(2)}) \\
 & \times p_{k-1}(M_{k-1} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k-1}^{(1)}, X_{0:k-1}^{(2)}) \delta M_{k-1}
 \end{aligned} \tag{4.17}$$

where equations (4.16) and (4.17) are augmented feature map update and prediction posteriors respectively. Their PHDs are recursively propagated using a PHD filter as follows.

4.4.1 PHD filter for Augmented Map posterior estimation

PHD prediction step

If PHD of the augmented feature map posterior with type β_{k-1} at time $k-1$ is given by,

$$D_{k-1|k-1}(\zeta_{k-1}, \beta_{k-1} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k-1}^{(1)}, X_{0:k-1}^{(2)}) \tag{4.18}$$

then, the PHD of the augmented feature map prediction distribution (4.17) corresponds to features with type β_k can be obtained as,

$$\begin{aligned}
 & D_{k|k-1}(\zeta_k, \beta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\
 &= b_k(\zeta_k, \beta_k | X_k^{(1)}, X_k^{(2)}) \\
 &+ \sum_{\beta_{k-1}=0}^a \int P_S(\zeta_{k-1}, \beta_{k-1}) f_M(\zeta_k, \beta_k | \zeta_{k-1}, \beta_{k-1}, X_k^{(1)}, X_k^{(2)}) \\
 & \times D_{k-1|k-1}(\zeta_{k-1}, \beta_{k-1} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k-1}^{(1)}, X_{0:k-1}^{(2)}) d\zeta_{k-1}
 \end{aligned} \tag{4.19}$$

In here $b_k(\zeta_k, \beta_k | X_k^{(1)}, X_k^{(2)})$ denotes the intensity of the newly appearing features of type β_k in the joint sensor FOV, which can be written using Campbell's theorem

(for marked point processes) [95] as follows,

$$b_k(\zeta_k, \beta_k | X_k^{(1)}, X_k^{(2)}) = \pi_k(\beta_k) \tilde{b}_k(\zeta_k | \beta_k, X_k^{(1)}, X_k^{(2)}) \quad (4.20)$$

where $\pi_k(\beta_k)$ denotes the birth intensity and $\tilde{b}_k(\zeta_k | \beta_k, X_k^{(1)}, X_k^{(2)})$ denotes the distribution of birth state, while a denotes the total number of feature types (having unique motion models). Total number of features (both static and dynamic) in the predicted augmented feature map can be obtained by,

$$N_{k|k-1} = \int D_{k|k-1}(\zeta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) d\zeta_k \quad (4.21)$$

PHD update step

Assuming that the number of false features (clutter) produced by the exteroceptive sensor mounted on the r th vehicle ($r = 1, 2$) is Poisson distributed with an average of $\lambda_c^{(r)}$, and their physical distribution given by $c_k^{(r)}(z_k^{(r)})$ is uniform over its sensor FOV; then PHD of the augmented feature map update posterior distribution (4.13) of features with type β_k can be obtained by adopting the "General multi-sensor update method" (Chapter 3) as follows,

$$\begin{aligned} D_{k|k}(\zeta_k, \beta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) &= (1 - P_D^{(1)})(1 - P_D^{(2)}) D_{k|k-1}(\zeta_k, \beta_k) \\ &+ \left[\sum_{\mathcal{P} \boxtimes_2 Z_k} \omega_{\mathcal{P}} \sum_{W \in \mathcal{P}} \rho_W(\zeta_k) \right] D_{k|k-1}(\zeta_k, \beta_k) \end{aligned} \quad (4.22)$$

where the abbreviation $D_{k|k-1}(\zeta_k, \beta_k)$ represents the PHD of the predicted feature map posterior (4.19) and $P_D^{(r)}$ represents the detection probability of the sensor mounted on the r th vehicle given by,

$$P_D^{(r)} = P_D^{(r)}(\zeta_k, \beta_k | X_k^{(r)}) \quad (4.23)$$

and the summation in (4.22) is taken over all so called "binary partitions" $\mathcal{P}$ of $Z_k = Z_k^{(1)} \cup Z_k^{(2)}$ (see [93] for more information). The notation, " $\mathcal{P} \boxplus_2 Z_k$ " stands for " $\mathcal{P}$ partitions Z_k into binary cells W ", where $W \in \mathcal{P}$ has one of the following forms,

$$W = \{z_k^{(1)}\}, \quad W = \{z_k^{(2)}\}, \quad W = \{z_k^{(1)}, z_k^{(2)}\} \quad (4.24)$$

The values of $\rho_W(\zeta_k)$ and $\omega_{\mathcal{P}}$ are given by,

$$\begin{aligned} \rho_W(\zeta_k) &= \begin{cases} \frac{P_D^{(1)} l_{z_k^{(1)}}^{(1)} (1 - P_D^{(2)})}{1 + D_{k|k-1} [P_D^{(1)} l_{z_k^{(1)}}^{(1)} (1 - P_D^{(2)})]} & \text{if } W = \{z_k^{(1)}\} \\ \frac{(1 - P_D^{(1)}) l_{z_k^{(2)}}^{(2)} P_D^{(2)}}{1 + D_{k|k-1} [(1 - P_D^{(1)}) l_{z_k^{(2)}}^{(2)} P_D^{(2)}]} & \text{if } W = \{z_k^{(2)}\} \\ \frac{P_D^{(1)} l_{z_k^{(1)}}^{(1)} P_D^{(2)} l_{z_k^{(2)}}^{(2)}}{D_{k|k-1} [P_D^{(1)} l_{z_k^{(1)}}^{(1)} P_D^{(2)} l_{z_k^{(2)}}^{(2)}]} & \text{if } W = \{z_k^{(1)}, z_k^{(2)}\} \end{cases} \end{aligned} \quad (4.25)$$

and,

$$\omega_{\mathcal{P}} = \frac{\prod_{W \in \mathcal{P}} d_W}{\sum_{\mathcal{Q} \boxplus_2 Z_k} \prod_{W \in \mathcal{Q}} d_W} \quad (4.26)$$

and,

$$d_W = \begin{cases} 1 + D_{k|k-1} [P_D^{(1)} l_{z_k^{(1)}}^{(1)} (1 - P_D^{(2)})] & \text{if } W = \{z_k^{(1)}\} \\ 1 + D_{k|k-1} [(1 - P_D^{(1)}) l_{z_k^{(2)}}^{(2)} P_D^{(2)}] & \text{if } W = \{z_k^{(2)}\} \\ D_{k|k-1} [P_D^{(1)} l_{z_k^{(1)}}^{(1)} P_D^{(2)} l_{z_k^{(2)}}^{(2)}] & \text{if } W = \{z_k^{(1)}, z_k^{(2)}\} \end{cases} \quad (4.27)$$

where for any function $h(\zeta_k, \beta_k)$, the abbreviation $D_{k|k-1}[h(\zeta_k, \beta_k)]$ is given by,

$$D_{k|k-1}[h(\zeta_k, \beta_k)] = \sum_{\beta_k=0}^a \int h(\zeta_k, \beta_k) D_{k|k-1}(\zeta_k, \beta_k) d\zeta_k \quad (4.28)$$

and

$$l_{z_k^{(1)}}^{(1)}(\zeta_k, \beta_k) = \frac{g_k^{(1)}(z_k^{(1)} | \zeta_k, \beta_k, X_k^{(1)})}{\lambda_c^{(1)} c_k^{(1)}(z_k^{(1)})} \quad (4.29)$$

$$l_{z_k^{(2)}}^{(2)}(\zeta_k, \beta_k) = \frac{g_k^{(2)}(z_k^{(2)} | \zeta_k, \beta_k, X_k^{(2)})}{\lambda_c^{(2)} c_k^{(2)}(z_k^{(2)})} \quad (4.30)$$

How GM-PHD filter works in CMSLAMMOT

We take a simple example (Fig. 4.2) in order to illustrate how PHD of the dynamic features evolve over time. Here, two vehicles perform CMSLAMMOT from time step k to $k + 5$. At time k , both vehicles observe a dynamic feature coming towards the vehicles and both vehicles acquire observations from it. From time $k + 1$, it can be seen that a Gaussian corresponds to the dynamic object emerges at the location where the dynamic feature is and moves along with it. It is also clear that, as time passes, the spread of the Gaussian reduces and the magnitude increases, which yields that over time, it is possible to infer the location of the dynamic object with high confidence. The kinematic state corresponds to this Gaussian, contains information such as velocity, heading angle etc. of the dynamic object.

4.4.2 Extraction of static and dynamic features

Now it is possible to extract the PHD component corresponds to any feature type β_k using the resultant PHD (4.22). We can obtain the PHD of the static feature map posterior from,

$$D_{k|k}(\zeta_k, \beta_k = 0 | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \quad (4.31)$$

and hence it is possible to extract the static feature locations and clearly it is possible to evaluate the CMSLAM posterior distribution (4.4). The PHD that corresponds

(a) PHD at time step k (b) PHD at time step $k+1$ (c) PHD at time step $k+2$ (d) PHD at time step $k+3$ (e) PHD at time step $k+4$ (f) PHD at time step $k+5$

Figure 4.2: How the PHD of the dynamic features evolves over time in an example CMSLAMMOT scenario. The yellow circles denote actual static landmark locations, while dotted lines denote the observations of each vehicle. A dynamic object emerges at $(15, -2.5)$ at time step k and moves towards the vehicles. It can be seen that over time, the Gaussian correspond to the dynamic object increases in magnitude, while its spread is reduced.

to the dynamic features can be obtained from,

$$\sum_{\beta_k=1}^a D_{k|k}(\zeta_k, \beta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \quad (4.32)$$

and the PHD of all feature types, i.e. PHD of the augmented feature map posterior (4.14) is given by,

$$\sum_{\beta_k=0}^a D_{k|k}(\zeta_k, \beta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \quad (4.33)$$

Once the PHD is known, actual locations of features of any type can be easily found [51] [73]. In order to calculate the particle weights (discussed in the section 4.5), we are required to evaluate the number of predicted and actual features present in the static feature map at time k . The number of predicted static features can be obtained as,

$$N_{k|k-1} = \int D_{k|k-1}(\zeta_k, \beta_k = 0 | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \delta \zeta_k \quad (4.34)$$

and the number of actual static features after measurement update can be obtained as,

$$N_{k|k} = \int D_{k|k}(\zeta_k, \beta_k = 0 | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \delta \zeta_k \quad (4.35)$$

The PHD filter prediction (4.19) and update (4.22) equations are evaluated using a GM-PHD filter [51].

4.5 Vehicle Trajectories Posterior Estimation

Instead of using both static and dynamic features (using augmented feature map), we propagate the joint vehicle trajectories posterior only along with the static feature posterior (4.4). The main reason behind this is that, the estimation of dynamic

feature locations could be slightly less accurate than the estimation of static feature locations. Hence an improved localization accuracy can be achieved by using static feature map alone.

By far, particle filter based vehicle trajectory estimation method introduced in FastSLAM [20] can be considered as the most suitable approach for practical SLAM solutions due to the adoptability of complex non-linear motion models in highly dynamic, unstructured and uncertain environmental conditions. In the same spirit, we propose propagating the vehicle trajectory posterior using a particle filter as follows.

The joint vehicle trajectories posterior at time k is given by,

$$p_{k|k}(X_{1:k}^{(1)}, X_{1:k}^{(2)} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \quad (4.36)$$

which can be represented by a set of weighted particles denoted by Ω_k as follows,

$$\Omega_k = \left\{ w_k^{[i]}, X_{1:k}^{(1),[i]}, X_{1:k}^{(2),[i]} \right\}_{i=1}^{N_s} \quad (4.37)$$

where, $w_k^{[i]}$ is the weight of i th particle, while N_s denotes the number of particles. Similar to FastSLAM, the particle set is recursively propagated in time, using sequential importance re-sampling (SIR) approach as follows.

Assume that the joint vehicle trajectories posterior at time $k - 1$ is represented by the set of weighted particles Ω_{k-1} . Then at time k , a new joint vehicle pose is sampled from each particle as follows,

$$(X_k^{(1),[i]}, X_k^{(2),[i]}) \sim f_X(X_k^{(1)}, X_k^{(2)} | X_{k-1}^{(1),[i]}, X_{k-1}^{(2),[i]}, U_k^{(1)}, U_k^{(2)}) \quad (4.38)$$

Although (4.38) is given as a joint transition, due to the conditional independence of each vehicle's motion, individual vehicle pose can be sampled using the correspond-

ing motion model. The new joint vehicle pose, $(X_k^{(1),[i]}, X_k^{(2),[i]})$, is then added to the set of particles Ω_{k-1} , creating a temporary set of particles distributed according to the proposal distribution given by,

$$q_{k|k}(X_{1:k}^{(1),[i]}, X_{1:k}^{(2),[i]} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \quad (4.39)$$

Now, each particle in the temporary set is assigned an importance weight given by,

$$\begin{aligned} w_k^{[i]} &= \frac{\text{target distribution}}{\text{proposal distribution}} \\ &= \frac{p_{k|k}(X_{1:k}^{(1),[i]}, X_{1:k}^{(2),[i]} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)})}{q_{k|k}(X_{1:k}^{(1),[i]}, X_{1:k}^{(2),[i]} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)})} \end{aligned} \quad (4.40)$$

Ignoring the normalization constant and factorizing the target and proposal distributions, it can be shown that,

$$w_k^{[i]} \propto l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}) \quad (4.41)$$

The value of $l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]})$ can be obtained using the same factorization as in (1.16) on the static feature map posterior,

$$p_{k|k}(S_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{1:k}^{(1)}, X_{1:k}^{(2)}) \quad (4.42)$$

by using the approach proposed by Mullane et al. in [72] [73] as follows,

$$\begin{aligned} l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) &= g_k(Z_k^{(1)}, Z_k^{(2)} | S_k, X_k^{(1)}, X_k^{(2)}) \\ &\quad \times \frac{p_{k|k-1}(S_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})}{p_{k|k}(S_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})} \end{aligned} \quad (4.43)$$

where the equation (4.43) is true for all values of S_k . Therefore without loss of

generality, by assigning $S_k = \emptyset$,

$$\begin{aligned} l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) &= g_k(Z_k^{(1)}, Z_k^{(2)} | \phi, X_k^{(1)}, X_k^{(2)}) \\ &\times \frac{p_{k|k-1}(\phi | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})}{p_{k|k}(\phi | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})} \end{aligned} \quad (4.44)$$

Now since the static map is empty, we should consider all the received measurements as false alarms (clutter). In addition, it is also assumed that these false features are independently and identically distributed over each vehicle's sensor FOV with the number of false features being Poisson distributed. Assuming that the average number of elements in S_k , is Poisson distributed, we can re-write (1.44) as follows,

$$\begin{aligned} l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}) \\ = \prod_{z_k^{(1)} \in Z_k^{(1)}} \lambda_c^{(1)} c_k^{(1)}(z_k^{(1)}) \prod_{z_k^{(2)} \in Z_k^{(2)}} \lambda_c^{(2)} c_k^{(2)}(z_k^{(2)}) \\ \times \exp(N_{k|k}^{[i]} - N_{k|k-1}^{[i]} - \lambda_c^{(1)} - \lambda_c^{(2)}) \end{aligned} \quad (4.45)$$

The importance weight of each particle in the temporary set is normalized and a new set of N_s particles are drawn with replacement, where each particles is sampled with a probability proportional to it's importance weight. The resultant particle set with their importance weights, denoted by Ω_k , represents the joint vehicle trajectory posterior (4.36).

4.6 Results and Analysis

Simulation results

A simulation is conducted in order to evaluate the feasibility and performance of the proposed CMSLAMMOT algorithm. Two identical vehicles are driven on two dif-

ferent vehicle trajectories on a square area of 120mx120m, consisting of 57 randomly placed static features (Fig. 4.3). Five dynamic features having non-linear dynamic motion models are used to evaluate the moving object tracking (MOT) performance. The vehicular parameters are summarized in Table 4.1. It is assumed that the average number of false alarms produced by each sensor is Poisson distributed, and generate uniformly over the sensor FOV $[-\pi/2, \pi/2] \times [0, 30]$.

Three different motion models are used for the simulation. The motion model $\beta_k = 0$ is used to denote the static features having the motion model of dirac delta function. Motion models $\beta_k = 1$ and $\beta_k = 2$ denote two dynamic motion models having the following motion model,

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \cos(\theta_k + \Omega_k)V_k\Delta T \\ y_k + \sin(\theta_k + \Omega_k)V_k\Delta T \\ \theta_k + \Omega_k \end{bmatrix} \quad (4.46)$$

where, ΔT denotes the time step. Ω_k denotes the change of heading angle and V_k denotes the translational velocity, which are given by,

$$\Omega_k = \Omega + \vartheta_k \quad (4.47)$$

$$V_k = V + \nu_k \quad (4.48)$$

In here ϑ_k and ν_k represents zero mean Gaussian noise having standard deviations $\sigma_\vartheta = 0.5^\circ$ and $\sigma_\nu = 0.3 \text{ ms}^{-1}$ respectively. In both models, $\beta_k = 1$ and $\beta_k = 2$, V is set to 5 ms^{-1} while Ω is set at 2° and -2° respectively.

Trajectories of the dynamic features ($\beta_k = 1$ and $\beta_k = 2$) are generated using the motion model (4.46) and the switching between different motion models is modeled

by the Markovian transition probability matrix,

$$[t_{k|k-1}(\beta_k|\beta_{k-1})] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.6 & 0.4 \\ 0 & 0.4 & 0.6 \end{bmatrix} \quad (4.49)$$

and the distribution of the motion models at birth is given by,

$$[\pi_k(\beta_k)] = [.9 \ .05 \ .05] \quad (4.50)$$

Probability of survival of each feature is given by,

$$[P_S] = [0.99 \ 0.95 \ 0.95] \quad (4.51)$$

and the probability of detection of each feature type is given by,

$$[P_D] = [0.95 \ 0.93 \ 0.93] \quad (4.52)$$

Fig. 4.3, shows the simulation environment with the trajectories of the moving Targets. The ground truths of each collaborating vehicle is also shown along with a set of accumulated false feature measurements during a sample run. As shown in Fig. 4.4, starting from origin both vehicles perform CMSLAMMOT, while each moving target maneuvers in the sensor FOV. Target 1 moves across the FOV of the second vehicle, while target 2 starts within the FOV and then move outward. Target 3 move towards the first vehicle from outside the FOV. Target 4 and Target 5 start within the FOV of first and second vehicles respectively and cross their FOVs navigating through the overlapped area of FOVs.

Fig. 4.4 shows estimated vehicle trajectories (superimposed on their ground truths),

estimated static feature locations and trajectories of estimated dynamic features (superimposed on their actual trajectories) from a sample run. It is clear that the proposed CMSLAMMOT solution is capable of tracking moving objects present in the sensor FOV, while producing acceptable deviation of the estimated vehicle trajectories from the ground truth.

Fig.4.5 compares the RMS positional errors of each vehicle against the proposed RFS-CMSLAM-II solution (chapter 3). It is clear that the proposed CMSLAMMOT solution is capable of tracking dynamic features present in the sensor FOV without compromising the SLAM accuracy of the original solution. Both solutions produce almost identical RMS positional errors, which is due to the fact that the proposed CMSLAMMOT solution could obtain the static map of the environment filtering out the moving objects present in the sensor FOV. Moreover, the benchmarked CMSLAM solution considers the moving objects as clutter and are filtered out of the map, as a result, both solutions produce almost identical static feature maps yielding almost identical RMS positional errors. A comparison of the number of static features present in the map is shown in Fig.4.6. It is clear that both solutions produce almost identical amount of static features. Since the proposed CMSLAMMOT filters out all the moving objects when extracting the static features map, it is clear that the RFS-CMSLAM-II solution also is capable of filtering out the moving objects present in the sensor FOV, assuming it is clutter.

The map error is compared in terms of the OSPA distance [77] in Fig. 4.7. Although it is clear that the proposed CMSLAMMOT solution produces almost identical results to that of the benchmarked RFS-CMSLAM-II solution, the RFS-CMSLAM-II solution marginally outperforms the CMSLAMMOT solution.

Figure 4.3: Ground truth of the vehicle trajectories (vehicle 1 in blue and the vehicle 2 in black) with randomly placed static features superimposed on the set of accumulated false measurements (in green) recorded during a sample run under the clutter rate $\lambda_c^r = 1$, where $r = 1, 2$ denotes the vehicle number. The tracks of the dynamic features represent in cyan. The number of static features present in the environment is 57 and the number of dynamic features used is 5.

(a) Vehicle 1

Figure 4.4: A comparison of estimated vehicle trajectories (first vehicle in green and second in red) superimposed on the ground truth (first vehicle in blue and second in black). Actual static feature locations are shown in black stars and the corresponding estimates are shown in cyan circles. Actual trajectories of the dynamic features are shown in yellow and the corresponding estimates are shown in magenta circles.

(a) CMSLAMMOT

(b) CMSLAM

Figure 4.5: RMS Vehicle position errors from 15 MC runs (first vehicle in green and the second in red).

Figure 4.6: A comparison of average number of static features present in the map. It is clear that the proposed CMSLAMMOT algorithm produces almost identical results to that of the CMSLAM solution.

Figure 4.7: A comparison of average map error in terms of OSPA distance. It can be seen that the CMSLAMMOT algorithm produces slightly larger map errors compared to the CMSLAM algorithm.

Figure 4.8: A picture taken during the experiment. The pillars present in the picture were used as static features and two pedestrians were used as dynamic features present in the environment. This figure shows two pedestrians crossing the overlapped sensor FOVs at the same time.

Table 4.1: Parameters Used in the Simulation (Multi-Vehicle)

Vehicle Parameters		Values
Velocity	V	2m/s
Sensor FOV	Range (r)	0 - 30m
	Bearing (b)	$-\pi/2$ - $+\pi/2$
Control Noise	Velocity (σ_v)	0.2m/s
	Steering Angle (σ_a)	2^0
Measurement Noise	Range (σ_r)	0.1m
	Bearing (σ_b)	1^0
Clutter rate	$\lambda_c^{(r)}$	1

Table 4.2: Parameters Used in the Simulation (Single-Vehicle)

Vehicle Parameters		Values
Velocity	V	4m/s
Sensor FOV	Range (r)	0 - 120m
	Bearing (b)	$-\pi/2$ - $+\pi/2$
Control Noise	Velocity (σ_v)	0.3m/s
	Steering Angle (σ_a)	2^0
Measurement Noise	Range (σ_r)	0.3m
	Bearing (σ_b)	0.5^0

RFS Single Vehicle SLAM with Moving Object tracking

For completeness simulations are also conducted with a single vehicle RFS SLAM (RFS-MONO-SLAM) in order to evaluate the feasibility and performance of the proposed MOT approach (SLAMMOT). A simulation environment consisting of 35 randomly placed landmarks are chosen along with three dynamic features (Fig.4.10(b)). The vehicular parameters used in the simulation are summarized in Table 4.2. The vehicle is driven on a preplanned trajectory while three targets starting at different times were moved according to their individual motion models. These targets are tracked while they are in the sensor FOV of the vehicle as seen in the simulation results. Simulations are conducted under low and high clutter conditions.

Switching between different motion models by each target is given by the Markovian

Figure 4.9: Estimated vehicle trajectories (first vehicle in green and the second in red) superimposed on the known way-points (first vehicle in blue and the second in red) along with estimated static feature locations (cyan). Estimated dynamic feature trajectories are shown in magenta and the corresponding actual trajectories (approximately) is shown in black.

Figure 4.10: Single vehicle RFS SLAM with MOT (SLAMMOT). A comparison of estimated vehicle trajectories (in red) and estimated moving target trajectories (blue circles) superimposed on ground truth of vehicle trajectories (in blue) and ground truth of moving target trajectories (in black) with estimated features (red circles) and actual features (green stars).

Figure 4.11: Single vehicle RFS SLAM with MOT. RMS Vehicle position error of 10 MC runs under low (in green) and high (in red) clutter conditions.

Figure 4.12: Average map error (static features) during RFS SLAMMOT in terms of OSPA distance under low (in green) and high (in red) clutter conditions.

Figure 4.13: Average number of static features present in the map during the SLAM-MOT under low (in green) and high (in red) clutter conditions. The actual number of static features observed is shown in black

transition probability matrix,

$$[t_{k|k-1}(\beta_k|\beta_{k-1})] = \begin{bmatrix} 0.99990 & 0.00005 & 0.00005 \\ 0.00002 & 0.59999 & 0.39999 \\ 0.00002 & 0.59999 & 0.39999 \end{bmatrix} \quad (4.53)$$

and the distribution of the motion models at birth is given by,

$$[\pi_k(\beta_k)] = [.99 \ .005 \ .005] \quad (4.54)$$

Probability of survival of each feature type given by,

$$P_S(\beta_k) = [0.99 \ 0.95 \ 0.95] \quad (4.55)$$

which can take different values depending on the feature type. In practice the probability of survival of a static feature is higher than that of a dynamic feature

due to highly uncertain motion models. Probability of detection of each feature type is given by,

$$P_D(\beta_k) = [0.99 \ 0.9 \ 0.9] \quad (4.56)$$

The estimated vehicle trajectory and the ground truth is compared in Fig.4.11. In Fig.4.12, quality of the static map is compared in terms of OSPA distance. The number of static features observed over time is compared in Fig. 4.13. It is clear that the SLAM with MOT solution is capable of extracting the static features from a combination of both static and dynamic features.

Experimental Results

Two robots each equipped with a laser scanner and a fiber optic gyroscope are used in the experiment. Velocity of each robot is measured using a built-in wheel encoder. Starting from the same location, the robots are simultaneously driven on two pre-planned trajectories (with known waypoints). The pillars present in the experimental area are used as static features and two pedestrians (with reflectors to improve the feature detection) are used to perform various maneuvers across, along and towards the joint sensor FOV, and CMSLAMMOT is performed using the acquired sensor data. We adopt the same non-holonomic dynamic motion equations as shown in (4.46) with changing the current heading angles by 2° ($\beta_k = 1$) and -2° ($\beta_k = 2$) depending on the motion model to describe the pedestrian motion. Switching between different motion models is described with the same transition matrix as in (4.49). The pedestrians were asked to maneuver with almost same speed during the experiment and the initial values set for V is 1.

The Fig.4.8 shows the experiment area and the two robots perform CMSLAMMOT, while two pedestrians are crossing the joint sensor FOV from two different directions

at the same time. Average speed of each vehicle is approximately 0.4 m/s and the range of observations are limited to 10m with the bearing range from 90° to 90° . Number of particles used for the estimation of vehicle trajectories is 100. Fig.4.9 shows the estimated vehicle trajectories, estimated landmark locations and estimated dynamic feature trajectories.

It is clear from the experimental results that the proposed CMSLAMMOT solution produces adequately accurate joint vehicle trajectory estimates, while estimating the trajectories of the dynamic features present in the joint sensor FOV. It can also be seen that the proposed solution is cable of tracking multiple dynamic features present in the joint sensor FOV at the same time. Further performance improvements in terms of MOT can be achieved by using more sophisticated motion models.

4.7 Conclusion

In this chapter we have presented a novel solution to the Collaborative Multi-Vehicle Simultaneous Localization and Mapping with Moving Object Tracking (CMSLAM-MOT) problem by extending our random finite set (RFS) based CMSLAM solution.

Unlike the RFS CMSLAM solution proposed in 3, we have broaden the usual definition of state of a feature by appending the type in addition to its kinematic state to cater for detection and tracking of both static and dynamic features. Each feature may change its type and hence the motion model over time making various complex maneuvers. An augmented RFS is used to represent the union of static and dynamic features, and using FISST techniques we devise a Bayesian solution to propagate both static and dynamic feature maps (conditioned on the vehicle trajectories) from measurements with clutter.

Although both static and dynamic feature maps are available, the CMSLAM solu-

tion is devised only using static features as the static feature estimates are usually much more accurate than the dynamic feature estimates. Similar to the original RFS CMSLAM solution, we proceed by decomposing the CMSLAM posterior in to a product of the static feature map posterior conditioned on the history of vehicle trajectories and the vehicle trajectories posterior. The vehicle trajectories posterior is propagated using a particle filter and the resultant static feature map posterior is used in the weight update step.

The proposed solution inherits inbuilt map management, data association and clutter filtering, while accommodating for feature detection and feature survival uncertainties. Performance of the proposed algorithm is demonstrated via a simulation and an actual experiment.

Chapter 5

A Hierarchical Solution to the Collaborative Multi-Vehicle SLAM Problem

5.1 Introduction

In the proposed CMSLAM solution presented in Chapter 3, control commands and measurements from all collaborating vehicles are communicated into a powerful dedicated fusion node at each time step in order to compute the CMSLAM posterior. Once fused, resultant vehicle trajectories and the landmark map can be communicated back into the collaborating vehicles in order to be used in the high level applications. Although such an approach solves the CMSLAM problem in a statistically principled manner, in practice, centralized solutions introduce several drawbacks including exponentially increasing computation and communication requirements, and communication delays and drop-outs. The other two solution architectures that can be considered for solving the CMSLAM problem are hierarchical and dis-

tributed [94]. In a typical hierarchical solution, each participating robot performs mono-SLAM and produce a local sub-map of its vicinity [24]. The resultant sub-maps can be periodically fused into a single global map by appropriately aligning in order to minimize estimation errors. Once the fusion process is over, each robot restarts the mono-SLAM process by resetting their local map.

In [80] Williams et al. introduced one of the most notable hierarchical solutions to the Multi-Vehicle SLAM problem based on their original work on constrained local sub-map filtering (CLSF) [24] [23]. In this solution, each collaborating robot produces an independent sub-map, which is periodically fused into the global map by appropriately formulating constraints on common features found in the overlapped areas. Although this solution provides a scalable hierarchical solution for Multi-Vehicle SLAM problem, it stems from the EKF-SLAM solution [18]. As a result, it inherits all the drawbacks of the conventional SLAM solutions. This includes solving additional sub-problems such as data association, clutter filtering and landmark map management during the sub-mapping process. Moreover, ideal landmark detection and data association can create drawbacks in practical applications. More recently, Estrada et al. [104] proposed a Graph SLAM based Hierarchical Multi-Robot SLAM solution. The proposed solution is based on the fusion of regional maps using cycle constraints in order minimize the estimation errors and extract the global map. Due to the expensive detection of graph fusion and incremental minimum cycle basis algorithm, this solution is not extendable to a large number of robots in practice.

In order to overcome such issues, in this Chapter, we propose a novel hierarchical solution to the CMSLAM problem by introducing a random finite set (RFS) mono-SLAM [73] based sub-mapping process. Instead of communicating control commands and measurement data from collaborating vehicles at each time step, the resultant sub-maps along with the sequence of applied control commands due

to the sub-mapping process are periodically communicated into a powerful centralized fusion node, where a higher level Hierarchical Collaborative Multi-Vehicle SLAM (HCMSLAM) algorithm is run. Using the sub-maps and applied control commands as inputs, the joint vehicle trajectories posterior and the global landmark map posterior is evaluated. The resultant global landmark map and the joint vehicle trajectories can be communicated back into the collaborating vehicles to be used in higher level applications.

Section 5.2 formulates the proposed random finite set HCMSLAM solution. The system architecture of the proposed HCMSLAM solution is presented in section 5.3. In section 5.4, RFS formulation of measurement and landmark map models is discussed. Section 5.5 presents the global landmark map posterior estimation. Implementation details including the joint vehicle trajectories posterior estimation method are presented in Section 5.6. Section 5.7 presents the simulation results and section 5.8 concludes the chapter.

5.2 Formulation of Probabilistic Random Finite Set HCMSLAM solution

Unlike the proposed centralized RFS CMSLAM solution, in HCMSLAM, each collaborating vehicle performs RFS mono-SLAM and produces a local sub-map and a vehicle trajectory. Periodically, the resultant sub-map and the applied control commands at each vehicle are communicated into the centralized fusion node and fused. In order to illustrate this, let's consider a set of collaborative vehicles performing RFS mono-SLAM [73] during the time period 1 to l .

Let $U_{1:l}^{(r)} = [U_1^{(r)}, U_2^{(r)}, \dots, U_l^{(r)}]^T$ denote the time sequence of control commands ap-

plied to the vehicle r up to time l , where $U_l^{(r)}$ denotes the control command applied at time l . Let the time sequence of pose history of each vehicle be denoted by $Y_{1:l}^{(r)} = [Y_1^{(r)}, Y_2^{(r)}, \dots, Y_l^{(r)}]^T$, where $Y_l^{(r)}$ denotes the pose of vehicle r with respect to the local frame of reference, at time l . Let the time sequence of sets of measurements obtained using exteroceptive sensors mounted on each vehicle be denoted by $Z_{1:l}^{(r)} = [Z_1^{(r)}, Z_2^{(r)}, \dots, Z_l^{(r)}]$, where $Z_l^{(r)} = \{z_{l,1}^{(r)}, z_{l,2}^{(r)}, \dots, z_{l,n_l^{(r)}}^{(r)}\}$ denotes the measurement set received at time l , from vehicle r , while $n_l^{(r)}$ denotes the number of measurements. Let the landmark map (sub-map) at time l be denoted by the set $M_l = \{m_{l,1}, m_{l,2}, \dots, m_{l,\alpha_k}\}$, where α_k denotes the number of features available in the map at time l and each m_l denotes the state of a feature. Without loss of generality, let's consider the case $r = 1, 2$ to make the exposition clearer.

Starting from a known initial position $Y_0^{(r)}$, each collaborating vehicle performs RFS mono-SLAM [72] [73] and evaluates the posterior,

$$p_{l|l}(M_l^{(r)}, Y_{1:l}^{(r)} | Z_{1:l}^{(r)}, U_{1:l}^{(r)}, Y_0^{(r)}) \quad (5.1)$$

using the landmarks in the vicinity. The resultant local sub-map $M_l^{(r)}$ and the sequence of applied control commands $U_{1:l}^{(r)}$ in each collaborating vehicle are communicated into a centralized fusion node in order to propagate the HCMSLAM posterior given by,

$$p_{k|k}(M_k, X_{1:k}^{(1)}, X_{1:k}^{(2)} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \quad (5.2)$$

In here, each $X_0^{(r)}$ denotes the known initial position of vehicle r with respect to the global frame of reference. The RFS M_k denotes the global landmark map and $X_{1:k}^{(r)}$ denotes the history of vehicle poses with respect to the global frame of reference. Applied control commands at time k is denoted by $U_k^{(r)}$, which is derived from the

sequence of control commands applied during the sub-mapping process $U_{1:l}^{(r)}$. The observation set received from r th collaborating vehicle is denoted by $Z_k^{(r)}$. Note that in the proposed solution, these observations are derived from the resultant local sub-maps, instead of directly extracting from the sensor measurements. This reduces the amount of information being exchanged dramatically and hence reduces the work load at the fusion node. Moreover, this also results in a consistent global landmark map with a much more smaller number of false features.

We make the following assumptions in the proposed HCMSLAM solution.

- Initial vehicle positions are assumed to be known.
- Sufficient communication bandwidth is available at all times.

Although we do not address the initial vehicle position problem in the proposed solution, for practical applications it is possible to solve this issue by performing RFS mono-SLAM with moving object tracking (this is a special case of RFS CM-SLAMMOT solution proposed in Chapter 4 which can be easily adopted for a single vehicle case) in order to extract the positions of other collaborating vehicles. We assume that the communication bandwidth is sufficient and available at all times. The proposed solution is robust against communication delays between the fusion node and the collaborating vehicles since the actual processing time in the centralized node in general is comparatively smaller than the time taken to produce a local sub-map. Moreover, in order to proceed with the sub-mapping process, no fused results/information from the centralized fusion node are required at collaborating vehicles. As a result, communication delays do not degrade the performance of the proposed HCMSLAM solution.

5.3 Hierarchical Multi-Vehicle SLAM System Architecture

In this section we propose a hierarchical architecture in order to solve the CMSLAM problem. In a typical hierarchical solution each participating robot performs mono-SLAM and resultant sub-maps and vehicle positions can be fused at a higher level by identifying common landmarks between the sub-maps and aligning them to minimize the estimation errors. This has been the most common approach for sub-map fusion in EKF based multi-vehicle SLAM solutions due to the fact that they propagate the cross correlation between feature estimates. However in our solution, we combine both particle filter based vehicle trajectory posterior estimation and RFS based landmark map posterior estimation in order to derive a hierarchical solution to the CMSLAM problem.

In order to illustrate the concept, let's consider the case where a set of collaborative robots perform HCMSLAM in an unknown environment. Each collaborating vehicle performs RFS mono-SLAM by observing the features present in the vicinity and produce a local sub-map and a vehicle trajectory. Periodically, these resultant sub-maps and the sequences of applied control commands into vehicles are communicated into the higher level fusion node, where the proposed RFS HCMSLAM algorithm is run at a much lower frequency. These sub-maps and the sequences of applied control commands are respectively considered as observations and control inputs in the HCMSLAM algorithm.

Once the sub-map and the control commands are communicated, each collaborating vehicle resets its local sub-map and starts a new RFS mono-SLAM process considering the current vehicle position as the origin. Once the fusion process at the fusion node is over, resultant global landmark map and the updated origin of the local

Figure 5.1: RFS Single-Vehicle SLAM based sub-mapping process with two vehicles. At points A and C each vehicle initiates the RFS Single-Vehicle SLAM process with a new local frame of reference and reach points B and D respectively. Local sub-maps produced, along with the sequence of control commands are communicated into the CMSLAM algorithm. This process repeats from points B and D. Black stars denote the observed landmarks during the process.

frame of reference with respect to the global frame of reference can be communicated back into the collaborating robots. Hence, each vehicle is able to transform its current local sub-map and the vehicle trajectory into the global frame of reference if required by the higher level processes. Note that the sub-mapping process at individual vehicle does not require any information/results back from the fusion node. Hence the proposed solution doesn't compromise the accuracy of the SLAM process due to communication delays. Any communication delay could only affect other higher level processes that rely on the global landmark map. Moreover, it is also clear that the proposed approach doesn't suffer from the data incest problem since the local sub-map and the vehicle trajectory is reset in each vehicle once the sub-map and the control commands are communicated into the higher level fusion node. This procedure is graphically illustrated in Fig.5.2.

Figure 5.2: The HCMSLAM algorithm for a three vehicle scenario. Each participating vehicle produces a local sub-map and a vehicle trajectory, which are used as inputs in the upper level CMSLAM filter along with the sequence of applied control commands.

RFS mono-SLAM filter

Let's consider the time interval k to $k + 1$ of HCMSLAM algorithm. During this time interval, each collaborating vehicle r performs RFS mono-SLAM [72] [73] and evaluates the familiar SLAM posterior $p_{l|l}(M_l^{(r)}, Y_{1:l}^{(r)} | Z_{1:l}^{(r)}, U_{1:l}^{(r)}, Y_0^{(r)})$, by factorizing as a product of the vehicle trajectory posterior and the landmark map posterior conditioned on the vehicle trajectory as follows.

$$p_{l|l}(M_l^{(r)}, Y_{1:l}^{(r)} | Z_{1:l}^{(r)}, U_{1:l}^{(r)}, Y_0^{(r)}) = p_{l|l}(M_l^{(r)} | Z_{1:l}^{(r)}, Y_0^{(r)}) \times p_{l|l}(Y_{1:l}^{(r)} | Z_{1:l}^{(r)}, U_{1:l}^{(r)}, Y_0^{(r)}) \quad (5.3)$$

where, $Y_0^{(r)}$ is the origin of the local frame of reference and the corresponding location with respect to the global frame of reference is $X_{k-1}^{(r)}$. The RFS $M_l^{(r)}$ represents the local sub-map and the RFS $Z_{1:l}^{(r)}$ represents the measurements received from the exteroceptive sensor mounted on the vehicle r . The transition models of both landmark map and the measurements are modeled using FISST and the landmark map posterior is evaluated using a GM implementation of a PHD filter [51]. The

vehicle trajectory posterior is propagated using a Rao-Blackwellised particle filter.

Then, the resultant PHD of the RFS $M_l^{(r)}$, which represents a portion of the whole environment, along with the sequence of control commands applied into each vehicle $U_{1:l}^{(r)}$ are communicated into the higher level HCMSLAM algorithm to be used as observations and control commands. Note that due to the RFS mono-SLAM based sub-mapping process, the observations (sub-maps) at the HCMSLAM algorithm includes zero or much lower number of false features.

RFS HCMSLAM filter

At the end of time step k , each collaborating vehicle communicates the local sub-map $M_l^{(r)}$ and the sequence of applied control commands $U_{1:l}^{(r)}$ recorded during the time interval $k - 1$ to k into the centralized fusion node. Now we are required to evaluate the HCMSLAM posterior distribution (5.2) using this information as inputs. Unlike in the proposed centralized RFS CMSLAM solution (see chapter 3), in (5.2), the observation RFS of each vehicle $Z_k^{(r)}$ (for $r = 1, 2$) represent landmark locations (in the form of observations) observed during the RFS mono-SLAM based sub-mapping process. Observations in the HCMSLAM algorithm are derived from the resultant RFSs $M_l^{(1)}$ and $M_l^{(2)}$ using the following transformation,

$$Z_k^{(1)} = T(M_l^{(1)}, X_{k-1}^{(1)}) \quad (5.4)$$

$$Z_k^{(2)} = T(M_l^{(2)}, X_{k-1}^{(2)}) \quad (5.5)$$

where the landmark locations in the local sub-maps are converted into the global frame of reference using the estimated origin of the local frame of reference initiated at time $k - 1$. Note that the estimates of the RFSs $M_l^{(1)}$ and $M_l^{(2)}$ are independent of the estimated vehicle trajectories in the global frame of reference, yielding an

independent observation set at each time a sub-map is produced. The control commands used in the HCMSLAM algorithm are derived from the sequence of control commands applied into each vehicle during the RFS mono-SLAM process as,

$$U_k^{(1)} = U_{1:l}^{(1)} \quad (5.6)$$

$$U_k^{(2)} = U_{1:l}^{(2)} \quad (5.7)$$

Now we can proceed with factorizing the HCMSLAM posterior (1.2) into a product of the vehicle trajectories posterior and the global landmark map posterior conditioned on the vehicle trajectories as follows.

$$\begin{aligned} & p_{k|k}(M_k, X_{1:k}^{(1)}, X_{1:k}^{(2)} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \\ &= p_{k|k}(M_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\ &\times p_{k|k}(X_{1:k}^{(1)}, X_{1:k}^{(2)} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \end{aligned} \quad (5.8)$$

This factorization allows propagation of the global landmark map posterior and the joint vehicle trajectories posterior as two separate estimation problems. The global landmark map and the measurements are modeled as RFSs and their dynamics are modeled using FISST. The global landmark map posterior is propagated using a Gaussian mixture PHD (GM-PHD) filter [51] and the joint vehicle trajectories are propagated using a Rao-Blackwellised particle filter.

The proposed HCMSLAM algorithm is shown as a block diagram in Fig. 5.3. Each collaborating vehicle performs RFS Single Vehicle SLAM and produces a sub-map of the environment in its vicinity. Such sub-maps along with the applied control commands at each vehicle are communicated into a centralized fusion node, where the HCMSLAM algorithm is run. The HCMSLAM algorithm, fuses all the information that contains in the sub-maps and produces a single global map of the environment

Figure 5.3: Proposed HCMSLAM solution. Each collaborating robot performs RFS-MONO-SLAM and produce a vehicle trajectory and a sub-map. The sub-map along with the applied control commands applied during the sub-mapping process are communicated in to the fusion node. At the fusion node, a hierarchical RFS CMSLAM algorithm is run, which fuses all such information from all the collaborative robots and produces a single global map and estimate the global vehicle trajectories of the collaborating robots.

and estimate vehicle trajectories with respect to this map. A GM-PHD filter [51] is used to fuse the sub-maps and a particle filter with stratified resampling [97] is used to estimate the global vehicle trajectories.

5.4 RFS formulation of Measurement and Landmark Map Models

The measurement sets in the proposed HCMSLAM solution are derived from sub-maps resulting from RFS mono-SLAM based sub-mapping process. More specifically, unlike in the RFS CMSLAM solution (chapter 3), each collaborating vehicle periodically communicates the resultant sub-map into the centralized fusion node. These sub-maps are used as the measurement sets in the HCMSLAM algorithm.

A measurement set usually contains not only actual landmark generated measurements, but also false measurements (clutter). But, in our solution, due to the clutter filtering in RFS mono-SLAM based sub-mapping process, each sub-map contains comparatively a much smaller number of false landmarks. Hence, although we model each measurement set as a union of actual measurements and false measurements, the number of false measurements are usually much smaller than the proposed centralized RFS CMSLAM solution which uses raw sensor measurements. The ordering of these measurements carries no significance in solving the HCMSLAM problem. Hence the each measurement set can be naturally represented as a RFS, $Z_k^{(r)}$,

$$Z_k^{(r)} = \{z_{k,1}^{(r)}, z_{k,2}^{(r)}, \dots, z_{k,n_k}^{(r)}\} \in \mathcal{F}(\mathcal{Z}^{(r)}) \quad (5.9)$$

where $\mathcal{F}(\mathcal{Z}^{(r)})$ is the collection of all finite subsets of the measurement space $\mathcal{Z}^{(r)}$ corresponds to r th vehicle's sensor measurements. Note that unlike in the centralized RFS CMSLAM solution, in here, each $z_k^{(r)}$ represents a Cartesian coordinate instead of a range and bearing.

Due to the periodic fusion of sub-maps produced by collaborating robots, at each time step some of the landmarks existing in the global landmark map continue to exist with the new landmarks appearing in the joint FOV created by the area covered by the sub-maps. Moreover, some of the existing landmarks present in the global landmark map could disappear from the map as time passes. Hence, in order to model these requirements in a more natural and effective manner, the global landmark map is represented as the RFS, M_K ,

$$M_k = \{m_{k,1}, m_{k,2}, \dots, m_{k,l_k}\} \in \mathcal{F}(\mathcal{M}) \quad (5.10)$$

where each $m_k \in \mathcal{M}$ and $\mathcal{M}$ is the state space of the features. $\mathcal{F}(\mathcal{M})$ is the collection

all finite subsets of $\mathcal{M}$.

RFS Landmark Map Transition Model

Let M denote the unexplored unknown static map of the environment which is being explored by the collaborating vehicles, and let the RFS M_{k-1} denote the global landmark map at time $k - 1$ which is given by,

$$M_{k-1} = M \cap (ExtFoV(X_{0:k-1}^{(1)}) \cup ExtFoV(X_{0:k-1}^{(2)})) \quad (5.11)$$

where, $ExtFoV(X_{0:k-1}^{(r)}) = ExtFoV(X_0^{(r)}) \cup ExtFoV(X_1^{(r)}) \cup \dots \cup ExtFoV(X_{k-1}^{(r)})$, for $r = 1, 2$. Each $ExtFoV(X_k^{(r)})$ represents the set of features pass through the extended field of view (FoV) of each vehicle at time k . The value of $ExtFoV(X_k^{(r)})$ is obtained via the sub-mapping process (from time $k - 1$ to k) as, $ExtFoV(X_k^{(r)}) = FoV(Y_0^{(r)}) \cup FoV(Y_1^{(r)}) \dots \cup FoV(Y_l^{(r)})$, where $FoV(y)$ denotes the features acquired during the sub-mapping process using the exteroceptive sensor mounted on vehicle r at y (with respect to the local frame of reference). Since the local sub-mapping process restarts at each time step k in the HCMSLAM filter, it is clear that the sub-maps of the collaborating vehicles are reset, and the sub-mapping starts at a new origin, resulting an elegant hierarchical solution which avoids data incest problem.

Now the map transition density from time $k - 1$ to k can be written as,

$$M_k = M_{k-1} \cup \{(FoVExt(X_{0:k-1}^{(1)}) \cup FoVExt(X_{0:k-1}^{(2)})) \cap \overline{M}_{k-1}\} \quad (5.12)$$

where $\overline{M}_{k-1} = M - M_{k-1}$ is the set of features present in the environment but not in M_{k-1} . Hence the second term in (5.12) represents the newly appearing landmarks in the map at time k due to the sub-mapping process performed between time $k - 1$ and k . It is clear the landmark map at time k contains landmarks coming from time

$k - 1$ and the set of new landmarks entering the map due to sub-mapping process.

Let the RFS representing the global landmark map at time $k - 1$ be denoted by M_{k-1} , then the predicted global landmark map at time k is given by,

$$M_k = \Gamma_k(X_k^{(1)}, X_k^{(2)}) \cup \left[\bigcup_{\zeta_{k-1} \in M_{k-1}} \Upsilon(\zeta_{k-1}) \right] \quad (5.13)$$

where the RFS $\Gamma_k(X_k^{(1)}, X_k^{(2)})$ denotes the newly appearing landmarks due the sub-mapping process of each vehicle between time $k - 1$ and k , and the Bernoulli RFS $\Upsilon(\zeta_{k-1})$ denotes the predicted state of the landmark $\zeta_{k-1} \in M_{k-1}$, which takes the form $\{\zeta_k\}$ with probability of reappearance $P_S(\zeta_{k-1})$ or $\emptyset$ with probability $1 - P_S(\zeta_{k-1})$. Here, the value of P_S is clearly higher than that of the RFS mono-SLAM solution as a result of using a sub-map. The probability density of state transition of the landmark from ζ_{k-1} to ζ_k conditioned on the probability of survival can be written as $f_M(\zeta_k | \zeta_{k-1}, X_k^{(1)}, X_k^{(2)})$. The RFS $\Gamma_k(X_k^{(1)}, X_k^{(2)})$ is independent of the landmarks that are reappearing from the global landmark map M_{k-1} . The corresponding RFS map transition density [73] can be written as,

$$f_M(M_k | M_{k-1}, X_k^{(1)}, X_k^{(2)}) = \sum_{W \subseteq M_k} f_M(W | M_{k-1}) f_\Gamma(M_k - W | X_k^{(1)}, X_k^{(2)}) \quad (5.14)$$

where, $f_M(W | M_{k-1})$ is the transition density of the set of features existing in the map, M_{k-1} from time $k - 1$ to time k , while $f_\Gamma(M_k - W | X_k^{(1)}, X_k^{(2)})$ represents the density of the RFS $\Gamma_k(X_k^{(1)}, X_k^{(2)})$ of new landmarks appearing in the joint FOV created by the area covered by the sub-maps at time k .

RFS Landmark Measurement Model

Let M_k denote the global landmark map at time k , while $C_k^{(r)}$ denote the RFS representing the false measurements (clutter) present in the sub-map received from the r th vehicle. Then corresponding measurements can be represented by the RFS,

$$Z_k^{(r)} = C_k^{(r)} \cup \left[\bigcup_{\zeta_k \in M_k} \Theta_k^{(r)}(\zeta_k) \right] \quad (5.15)$$

where $\Theta_k^{(r)}(\zeta_k)$ is a Bernoulli RFS representing the measurement corresponds to the observation of landmark $\zeta_k \in M_k$. Due to the limited FOV (areas of the sub-maps), $\Theta_k^{(r)}(\zeta_k)$ can have a value of the form $\{z_k^{(r)}\}$ with probability of detection $P_D^{(r)}(\zeta_k | X_k^{(r)})$ or $\emptyset$ with a probability of $1 - P_D^{(r)}(\zeta_k | X_k^{(r)})$. Note that the elements present in the RFS $Z_k^{(r)}$ are specified as Cartesian coordinate values by transforming the landmark locations in the local sub-map using the origin of the local frame of reference in global frame of reference. Conditioned on detection, measurement likelihood function corresponds to the observation of landmark with state ζ_k by the r th vehicle is given by $g_k^{(r)}(z_k^{(r)} | \zeta_k, X_k^{(r)})$. The corresponding RFS measurement density can be written as,

$$g_k^{(r)}(Z_k^{(r)} | M_k, X_k^{(r)}) = \sum_{W \subseteq Z_k^{(r)}} g_Z^{(r)}(W | M_k, X_k^{(r)}) g_C^{(r)}(Z_k^{(r)} - W) \quad (5.16)$$

where, $g_Z^{(r)}(W | M_k, X_k^{(r)})$ denotes the density of actual feature observations, while $g_C^{(r)}(Z_k^{(r)} - W)$ denotes the density of clutter. Since the measurements are not directly obtained from a exteroceptive sensor, the number of false features are much smaller than that of during the sub-mapping process.

5.5 Global Landmark Map Posterior Estimation

In this section, we present global landmark map posterior propagation method. The map posterior is factorized into a recursive prediction/update steps and a Gaussian mixture (GM) implementation of a PHD filter [48] [51] is adopted in order to propagate the PHD of the global landmark map. A new PHD filter update method is also proposed by extending the general multi-sensor PHD update method based solution (RFS-CMSLAM-II) presented in the RFS CMSLAM solution (Chapter 3).

Let the global landmark map posterior conditioned on the vehicle trajectories at time k be denoted by,

$$p_{k|k}(M_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \quad (5.17)$$

then the corresponding PHD is given by,

$$D_{k|k}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) = \int p_{k|k}(M_k \cup \zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \delta M_k \quad (5.18)$$

and we can obtain the total number of landmarks in the global map in a region S can be obtained by,

$$N_{k|k} = \int_S D_{k|k}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) d\zeta_k \quad (5.19)$$

We proceed with factorizing the global landmark map posterior (5.17) as follows,

$$\begin{aligned} p_{k|k}(M_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) &= g_k(Z_k^{(1)}, Z_k^{(2)} | M_k, X_k^{(1)}, X_k^{(2)}) \\ &\times \frac{p_{k|k-1}(M_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})}{l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})} \end{aligned} \quad (5.20)$$

where the denominator in (5.20) is the normalization constant and,

$$\begin{aligned}
& p_{k|k-1}(M_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\
&= \int f_M(M_k | M_{k-1}, X_k^{(1)}, X_k^{(2)}) \\
&\times p_{k-1}(M_{k-1} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k-1}^{(1)}, X_{0:k-1}^{(2)}) \delta M_{k-1}
\end{aligned} \tag{5.21}$$

The equations (5.21) and (5.20) respectively correspond to the map prediction and update posteriors and the PHD of the global landmark map (5.17) can be obtained by recursively propagating their PHDs as follows.

5.5.1 PHD filter for Global map posterior estimation

PHD of the Global Landmark Map Prediction

The PHD of the global landmark map prediction posterior (5.21) can be obtained using the PHD of the map posterior at time $k-1$ and the map transition model as follows,

$$\begin{aligned}
& D_{k|k-1}(\zeta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\
&= b_k(\zeta_k | X_k^{(1)}, X_k^{(2)}) \\
&+ \int P_S(\zeta_{k-1}) f_M(\zeta_k | \zeta_{k-1}, X_k^{(1)}, X_k^{(2)}) \\
&\times D_{k-1|k-1}(\zeta_{k-1} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k-1}^{(1)}, X_{0:k-1}^{(2)}) d\zeta_{k-1}
\end{aligned} \tag{5.22}$$

where $b_k(\zeta_k | X_k^{(1)}, X_k^{(2)})$ denotes the intensity of newly appearing landmarks in the global map due to the local sub-mapping process. Total number of predicted landmarks in the map can be obtained by,

$$N_{k|k-1} = \int_S D_{k|k-1}(\zeta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) d\zeta_k \tag{5.23}$$

If the landmarks present in the global landmark map are assumed to be static and the probability of survival, $P_S(\zeta_{k-1})$, of landmark ζ_{k-1} in the global map from time step $k-1$ to k is considered constant the PHD of the landmark map update posterior can be simplified to,

$$\begin{aligned} D_{k|k-1}(\zeta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) &= b_k(\zeta_k | X_k^{(1)}, X_k^{(2)}) \\ &+ P_S D_{k-1|k-1}(\zeta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k-1}^{(1)}, X_{0:k-1}^{(2)}) \end{aligned} \quad (5.24)$$

PHD of the Global Landmark Map Update

We are required to update the PHD of the global landmark map using recently communicated sub-maps, which are adopted as observations. Unlike in centralized RFS CMSLAM solution, where observations are range, bearing values of features present in the sensor FOV, these sub-maps provide observations in the form of Cartesian coordinates of the features spanning over a larger area. Hence, in order to obtain the updated PHD of the global landmark map we propose to extend the PHD update technique introduced in the centralized RFS CMSLAM solution (chapter 3) as follows.

Assume that the number of false measurements (false landmarks) present in the local sub-map produced by the r th vehicle is Poisson distributed at an average of $\lambda_c^{(r)}$, and their physical distribution given by $c^{(r)}(z^{(r)})$. Then, PHD of the updated global landmark map can be obtained using the PHD of the predicted global landmark map as,

$$\begin{aligned} D_{k|k}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) &= (1 - P_D^{(1)})(1 - P_D^{(2)})D_{k|k-1}(\zeta_k) \\ &+ \left[\sum_{\mathcal{P} \in \Xi_2 Z_k} \omega_{\mathcal{P}} \sum_{W \in \mathcal{P}} \rho_W(\zeta_k) \right] D_{k|k-1}(\zeta_k) \end{aligned} \quad (5.25)$$

where the abbreviation $D_{k|k-1}(\zeta_k)$ represents the PHD of the predicted global landmark map posterior (5.21). The summation in (5.25) is taken over all so called "binary partitions" $\mathcal{P}$ of $Z_k = Z_k^{(1)} \cup Z_k^{(2)}$ (see [93] for more information). The notation, " $\mathcal{P} \boxplus_2 Z_k$ " stands for " $\mathcal{P}$ partitions Z_k into binary cells W ", where $W \in \mathcal{P}$ has one of the following forms,

$$W = \{z_k^{(1)}\}, \quad W = \{z_k^{(2)}\}, \quad W = \{z_k^{(1)}, z_k^{(2)}\} \quad (5.26)$$

The values of $\rho_W(\zeta_k)$ and $\omega_{\mathcal{P}}$ are given by,

$$\rho_W(\zeta_k) = \begin{cases} \frac{P_D^{(1)} l_{z^{(1)}}^{(1)}(\zeta_k)(1 - P_D^{(2)})}{1 + v_{k|k-1}[P_D^{(1)} l_{z^{(1)}}^{(1)}(1 - P_D^{(2)})]} & \text{if } W = \{z_k^{(1)}\} \\ \frac{(1 - P_D^{(1)}) l_{z^{(2)}}^{(2)}(\zeta_k) P_D^{(2)}}{1 + v_{k|k-1}[(1 - P_D^{(1)}) l_{z^{(2)}}^{(2)} P_D^{(2)}]} & \text{if } W = \{z_k^{(2)}\} \\ \frac{P_D^{(1)} l_{z^{(1)}}^{(1)}(\zeta_k) P_D^{(2)} l_{z^{(2)}}^{(2)}(\zeta_k)}{1 + v_{k|k-1}[P_D^{(1)} l_{z^{(1)}}^{(1)} P_D^{(2)} l_{z^{(2)}}^{(2)}]} & \text{if } W = \{z_k^{(1)}, z_k^{(2)}\} \end{cases} \quad (5.27)$$

and,

$$\omega_{\mathcal{P}} = \frac{\prod_{W \in \mathcal{P}} d_W}{\sum_{Q \boxplus_2 Z_k} \prod_{W \in Q} d_W} \quad (5.28)$$

where,

$$l_{z^{(1)}}^{(1)}(\zeta_k) = \frac{g_k^{(1)}(z_k^{(1)} | \zeta_k, X_k^{(1)})}{\lambda_k^{(1)} c_k^{(1)}(z_k^{(1)})} \quad (5.29)$$

$$l_{z_k^{(2)}}^{(2)}(\zeta_k) = \frac{g_k^{(2)}(z_k^{(2)} | \zeta_k, X_k^{(2)})}{\lambda_k^{(2)} c_k^{(2)}(z_k^{(2)})} \quad (5.30)$$

and,

$$d_W = \begin{cases} 1 + D_{k|k-1}[P_D^{(1)} l_{z^{(1)}}^{(1)}(1 - P_D^{(2)})] & \text{if } W = \{z_k^{(1)}\} \\ 1 + D_{k|k-1}[(1 - P_D^{(1)}) l_{z^{(1)}}^{(1)} P_D^{(2)}] & \text{if } W = \{z_k^{(2)}\} \\ 1 + D_{k|k-1}[P_D^{(1)} l_{z^{(1)}}^{(1)} P_D^{(2)} l_{z^{(2)}}^{(2)}] & \text{if } W = \{z_k^{(1)}, z_k^{(2)}\} \end{cases} \quad (5.31)$$

For any function $h()$, $D_{k|k-1}[h()]$ is given by,

$$D_{k|k-1}[h()] = \int h(\zeta_k) D_{k|k-1}(\zeta_k) d\zeta_k \quad (5.32)$$

The function $g_k^{(r)}(z^{(r)}|\zeta_k, X_k^{(r)})$ represents the measurement likelihood and $P_D^{(r)} = P_D^{(r)}(\zeta_k|X_k^{(r)})$ denotes the probability of detection of a landmark ζ_k present in the global map, by the r th vehicle during the RFS mono-SLAM based sub-mapping process and is considered as a constant.

5.6 Implementation of the HCMSLAM algorithm

In order to cater complex non-linear motion models, we adopt a particle filter based joint vehicle trajectory posterior propagation method as explained in this section.

In the case of two vehicles performing HCMSLAM, the joint vehicle trajectories posterior at time k is given by,

$$p_{k|k}(X_{1:k}^{(1)}, X_{1:k}^{(2)}|Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \quad (5.33)$$

The posterior (5.33) is represented by a set of weighted particles denoted by Ω_k and given by,

$$\Omega_k = \left\{ w_k^{[i]}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]} \right\}_{i=1}^{N_s} \quad (5.34)$$

where $w_k^{[i]}$ is the weight of the i th particle and N_s denotes number of particles used in the filter. Similar to FastSLAM [20], the vehicle trajectories posterior is propagated using a Rao-Blackwellised particle filter.

The global landmark map posterior (5.17) is conditioned on the joint vehicle trajectories. Hence if the joint vehicle trajectories are represented as a set of weighted

particles (5.34) and the global landmark map is represented in its PHD form (5.18), the joint global landmark map and vehicle trajectories posterior (i.e. Hierarchical CMSLAM posterior (5.2)) at time k can be represented as,

$$\left\{ w_k^{[i]}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}, D_{k|k}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}) \right\}_{i=1}^{N_s} \quad (5.35)$$

where, N_s is the number of particles, while $w_k^{[i]}$ denotes the importance weight of the i th particle. As it is clear, each particle contains a PHD of the map, conditioned on its joint vehicle trajectory.

The joint vehicle trajectories are propagated using a particle filter. Note that the number of particles to be used in the HCMSLAM solution should be much larger than the number of particles used in the centralized RFS CMSLAM solution. This is due to the prediction of vehicle positions for a sequence of time steps instead of a single time step. The PHD of the global landmark map is propagated using a Gaussian mixture implementation of a PHD filter (GM-PHD) [51].

5.6.1 Particle filter Implementation

Assume that the set of weighted particles Ω_{k-1} represent the joint vehicle trajectories posterior at time $k-1$. Then at time k , a new joint vehicle pose is sampled from each particle as follows,

$$(X_k^{(1),[i]}, X_k^{(2),[i]}) \sim f_X(X_k^{(1)}, X_k^{(2)} | X_{k-1}^{(1),[i]}, X_{k-1}^{(2),[i]}, U_k^{(1)}, U_k^{(2)}) \quad (5.36)$$

Now, substituting from equations (5.6) and (5.7), we can rewrite (5.36) as,

$$(X_k^{(1),[i]}, X_k^{(2),[i]}) \sim f_X(X_k^{(1)}, X_k^{(2)} | X_{k-1}^{(1),[i]}, X_{k-1}^{(2),[i]}, U_{1:l}^{(1)}, U_{1:l}^{(2)}) \quad (5.37)$$

Note that, although (5.37) is given as a joint transition model, due to the conditional independence between two vehicles' motion, each vehicle pose can be sampled using their individual motion model as follows,

$$X_k^{(r),[i]} \sim f_X^{(r)}(X_k^{(r)} | X_{k-1}^{(r),[i]}, U_{1:l}^{(r)}) \quad (5.38)$$

The new vehicle poses, $(X_k^{(1),[i]}, X_k^{(2),[i]})$, are then added to the set of particles Ω_{k-1} , creating a temporary set of particles distributed according to the proposal distribution given as,

$$q_{k|k}(X_{1:k}^{(1),[i]}, X_{1:k}^{(2),[i]} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)}) \quad (5.39)$$

Hence importance weight of each particle in the temporary set of particles is given by,

$$\begin{aligned} w_k^{[i]} &= \frac{\text{target distribution}}{\text{proposal distribution}} \\ &= \frac{p_{k|k}(X_{1:k}^{(1),[i]}, X_{1:k}^{(2),[i]} | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)})}{q_{k|k}(X_{1:k}^{(1),[i]}, X_{1:k}^{(2),[i]} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, U_{1:k}^{(1)}, U_{1:k}^{(2)}, X_0^{(1)}, X_0^{(2)})} \end{aligned} \quad (5.40)$$

By factorizing the target distribution using Bayes rule and ignoring the normalization constant it can be shown that the particle weight is given by,

$$w_k^{[i]} \propto l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}) \quad (5.41)$$

where $l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]})$ is the normalization constant in (5.20), which can be obtained by rearranging the global landmark map update

posterior (5.20) as follows,

$$l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) = g_k(Z_k^{(1)}, Z_k^{(2)} | M_k, X_k^{(1)}, X_k^{(2)}) \times \frac{p_{k|k-1}(M_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})}{p_{k|k}(M_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})} \quad (5.42)$$

The equation (5.42) is true for all values of M_k . Therefore, without loss of generality, by assigning $M_k = \phi$ we can obtain,

$$l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) = g_k(Z_k^{(1)}, Z_k^{(2)} | \phi, X_k^{(1)}, X_k^{(2)}) \times \frac{p_{k|k-1}(\phi | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})}{p_{k|k}(\phi | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)})} \quad (5.43)$$

Assuming that the number of landmarks present in M_k is Poisson distributed, it can be shown that,

$$\begin{aligned} & l_{k|k-1}(Z_k^{(1)}, Z_k^{(2)} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) \\ &= \prod_{z^{(1)} \in Z_k^{(1)}} \lambda_c^{(1)} c^{(1)}(z^{(1)}) \prod_{z^{(2)} \in Z_k^{(2)}} \lambda_c^{(2)} c^{(2)}(z^{(2)}) \\ & \times \exp(N_{k|k} - N_{k|k-1} - \lambda_c^{(1)} - \lambda_c^{(2)}) \end{aligned} \quad (5.44)$$

where $N_{k|k-1}$ and N_k respectively denote the number of landmarks present in the global landmark map before and after the measurement update. Here, since the global map is empty, we consider all the measurements (features present in the sub-maps) as false alarms (clutter). The false measurements are assumed to be independently and identically distributed over on each sub-map with the number of false features being Poisson distributed.

The importance weight $w_k^{[i]}$ of each particle in the temporary set is normalized so that $\sum_{i=1}^{N_s} w_k^{[i]} = 1$. Then, a new set of N_s particles is created by drawing particles from the temporary with replacement, where each particle is drawn with a probability proportional to its importance weight. The resultant particle set with

their importance weights, denoted by Ω_k , represents the target posterior density (5.33).

5.6.2 GM-PHD filter implementation

Let the PHD of the global landmark map at time $k - 1$ be denoted by a mixture of Gaussians of the form,

$$D_{k-1|k-1}(\zeta_{k-1} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k-1}^{(1),[i]}, X_{0:k-1}^{(2),[i]}) = \sum_{j=1}^{J_{k-1}} \eta_{k-1}^{[i,j]} \mathcal{N}(\zeta_{k-1}; \mu_{k-1}^{[i,j]}, P_{k-1}^{[i,j]}) \quad (5.45)$$

where, J_{k-1} is the number of Gaussians, while $\eta_{k-1}^{[i,j]}$, $\mu_{k-1}^{[i,j]}$ and $P_{k-1}^{[i,j]}$ respectively denote weight, mean and covariance of the j^{th} Gaussian. Assume that the intensity of the new landmarks appearing in the joint area covered by the sub-maps at time k can also be represented as a mixture of Gaussians of the form,

$$b_k(\zeta_k | X_k^{(1),[i]}, X_k^{(2),[i]}) = \sum_{j=1}^{J_{b,k}} \eta_{b,k}^{[i,j]} \mathcal{N}(\zeta_k; \mu_{b,k}^{[i,j]}, P_{b,k}^{[i,j]}) \quad (5.46)$$

where, $J_{b,k}$ denotes the expected number of new landmarks. Then, the PHD of the predicted global landmark map posterior (5.22) simplifies into a mixture of Gaussians of the form,

$$D_{k|k-1}(\zeta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}) = \sum_{j=1}^{J_{k|k-1}} \eta_{k|k-1}^{[i,j]} \mathcal{N}(\zeta_{k|k-1}; \mu_{k|k-1}^{[i,j]}, P_{k|k-1}^{[i,j]}) \quad (5.47)$$

The measurement model is linearized using the linear Gaussian assumption, and the measurement likelihood function is written as,

$$g_k^{(r)}(z^{(r)} | \zeta_k, X_k^{(r)}) \approx \mathcal{N}(z^{(r)}; h_k^{(r)}(\zeta_k, X_k^{(r)}, 0), U_k^r R_k U_k^{rT}) \quad (5.48)$$

In equation (5.48), $U_k^{(r)}$ is the Jacobian of the measurement model given by the non-linear function $h_k^{(r)}(\zeta_k, X_k^{(r)}, \xi_k^{(r)})$ with respect to ξ_k at $\xi_k = 0$. The random vector $\xi_k^{(r)}$ denotes the zero-mean Gaussian noise with covariance $R_k^{(r)}$. Let, $H_k^{(r)}$ denote the Jacobian of $h_k^{(r)}(\zeta_k, X_k^{(r)}, 0)$ with respect to ζ_k at some $\zeta_k = \mu$. Then, we can show that the measurement updated PHD of the global landmark map posterior results in a mixture of Gaussians as follows.

The first term in the PHD update equation (5.21) produces a mixture of Gaussians due to (5.47). When W takes the value of $\{z_k^{(1)}\}$, substituting from (5.46) and (5.47), and using standard results of Gaussian functions [51] it can be shown that the value of $\rho_W D_{k|k-1}(\zeta_k)$ results in a mixture of Gaussians since,

$$\frac{P_D^{(1)} l_{z_k^{(1)}}^{(1)} (1 - P_D^{(2)})(\zeta_k) D_{k|k-1}(\zeta_k)}{1 + D_{k|k-1}[P_D^{(1)} l_{z_k^{(1)}}^{(1)} (1 - P_D^{(2)})]} = \sum_{j=1}^{J_{k|k-1}} \tilde{\eta}_{k|k}^{(1),[i,j]} \mathcal{N}(\zeta_k; \mu_{k|k}^{(1),[i,j]}, P_{k|k}^{(1),[i,j]}) \quad (5.49)$$

where,

$$\tilde{\eta}_{k|k}^{(1),[i,j]} = \frac{\eta_{k|k-1}^{[i,j]} P_D^{(1)} (1 - P_D^{(2)}) q_k^{(1),[i,j]}(z_k^{(1)}, X_k^{(1),[i]})}{\sum_{y=1}^{J_{k|k-1}} \eta_{k|k-1}^{[i,y]} P_D^{(1)} (1 - P_D^{(2)}) q_k^{(1),[i,y]}(z_k^{(1)}, X_k^{(1),[i]})} \quad (5.50)$$

the values of $\mu_{k|k}^{(r),[i,j]}$, $P_{k|k}^{(r),[i,j]}$ and $q_k^{(r),[i,j]}(z_k^{(r)}, X_k^{(r),[i]})$ are given by,

$$\mu_{k|k}^{(1),[i,j]} = \mu_{k|k-1}^{[i,j]} + K_k^{(1),[i,j]}(z_k^{(1)} - h_k^{(1)}(\mu_{k|k-1}^{[i,j]}, X_k^{(1),[i]}, 0)) \quad (5.51)$$

$$P_{k|k}^{(1),[i,j]} = (I - K_k^{(1),[i,j]} H_k^{(1),[i,j]}) P_{k|k-1}^{[i,j]} \quad (5.52)$$

$$q_k^{(1),[i,j]}(z_k^{(1)}) = \mathcal{N}(z_k^{(1)}; h_k^{(1)}(\mu_{k|k-1}^{[i,j]}, X_k^{(1),[i]}, 0), S_k^{(1),[i,j]}) \quad (5.53)$$

where $K_k^{(1),[i,j]}$ and $S_k^{(1),[i,j]}$ are given by,

$$K_k^{(1),[i,j]} = P_{k|k-1}^{[i,j]} H_k^{(1),[i,j]} (S_k^{(1),[i,j]})^{-1} \quad (5.54)$$

and

$$S_k^{(1),[i,j]} = U_k^{(1),[i,j]} R_k^{(1),[i,j]} U_k^{(1),[i,j]T} + H_k^{(1),[i,j]} P_{k|k-1}^{[i,j]} H_k^{(1),[i,j]T} \quad (5.55)$$

Similarly it can also be shown that the value of $\rho_W D_{k|k-1}(\zeta_k)$ results in a mixture of Gaussians when W takes the value of $\{z_k^{(2)}\}$.

When W is of the form $W = \{z_k^{(1)}, z_k^{(2)}\}$, $\rho_W(\zeta_k)$ results in a constant, while the multiplication between $\rho_W(\zeta_k)$ and $D_{k|k-1}(\zeta_k)$ results in a mixture of Gaussians as follows,

$$\rho_W(\zeta_k) D_{k|k-1}(\zeta_k) = \sum_{j=1}^{J_{k|k-1}} \tilde{\eta}_{k|k}^{(2),[i,j]} \mathcal{N}(\zeta_k; \mu_{k|k}^{(1,2),[i,j]}, P_{k|k}^{(1,2),[i,j]}) \quad (5.56)$$

where

$$\tilde{\eta}_{k|k}^{(2),[i,j]} = \frac{\eta_{k|k-1}^{[i,j]} P_D^{(1)} P_D^{(2)} q_k^{(1),[i,j]}(z_k^{(1)}, X_k^{(1),[i]}) q_k^{(1,2),[i,j]}(z_k^{(2)}, X_k^{(2),[i]})}{\sum_{y=1}^{J_{k|k-1}} \eta_{k|k-1}^{[i,y]} P_D^{(1)} P_D^{(2)} q_k^{(1),[i,y], X_k^{(1),[i]}}(z_k^{(1)}, X_k^{(1),[i]}) q_k^{(1,2),[i,y]}(z_k^{(2)}, X_k^{(2),[i]})} \quad (5.57)$$

where the values of $\mu_{k|k}^{(1,2),[i,j]}$, $P_{k|k}^{(1,2),[i,j]}$ and $q_k^{(1,2),[i,j]}(z_k^{(2)}, X_k^{(2),[i]})$ can be obtained as follows,

$$\mu_{k|k}^{(1,2),[i,j]} = \mu_{k|k-1}^{(1),[i,j]} + K_k^{(1,2),[i,j]}(z_k^{(2)} - h_k^{(2)}(\mu_{k|k-1}^{(1),[i,j]}, X_k^{(2),[i]}, 0)) \quad (5.58)$$

$$P_{k|k}^{(1,2),[i,j]} = (I - K_k^{(1,2),[i,j]} H_k^{(2),[i,j]}) P_{k|k}^{(1),[i,j]} \quad (5.59)$$

$$q_k^{(1,2),[i,j]}(z_k^{(2)}, X_k^{(2),[i]}) = \mathcal{N}(z_k^{(2)}; h_k^{(2)}(\mu_{k|k-1}^{(1),[i,j]}, X_k^{(2),[i]}, 0), S_k^{(1,2),[i,j]}) \quad (5.60)$$

where $K_k^{(1,2),[i,j]}$ and $S_k^{(1,2),[i,j]}$ are given by,

$$K_k^{(1,2),[i,j]} = P_{k|k}^{(1),[i,j]} H_k^{(2),[i,j]} (S_k^{(1,2),[i,j]})^{-1} \quad (5.61)$$

$$S_k^{(1,2),[i,j]} = U_k^{(2),[i,j]} R_k^{(2),[i,j]} U_k^{(2),[i,j]T} + H_k^{(2),[i,j]} P_{k|k}^{(1),[i,j]} H_k^{(2),[i,j]T} \quad (5.62)$$

Moreover it can also be shown that the value of $\omega_{\mathcal{P}}$ results in a positive real number less than 1. Therefore, it is apparent that the second term in the PHD update

equation (5.24) also produces a mixture of Gaussians of following form.

$$D_{k|k}(\zeta_k | Z_{1:k}^{(1)}, Z_{1:k}^{(2)}, X_{0:k}^{(1)}, X_{0:k}^{(2)}) = \sum_{j=1}^{J_{k|k}} \eta_{k|k}^{[i,j]} \mathcal{N}(\zeta_k; \mu_{k|k}^{[i,j]}, P_{k|k}^{[i,j]}) \quad (5.63)$$

5.7 Simulation Results

Performance of the proposed hierarchical CMSLAM (HCMSLAM) algorithm is evaluated using a simulation. Two vehicles with identical control and sensor parameters (summarized in Table.5.1) are driven on two different vehicle trajectories on a simulation environment consisting of 81 randomly placed landmarks. Each vehicle performs RFS mono-SLAM based sub-mapping process starting from a local frame of reference at the beginning of each time step of the RFS HCMSLAM algorithm. Resultant sub-map and the sequence of control commands applied by each vehicle during that time period are used as control commands and observations in the proposed RFS HCMSLAM solution.

It is assumed that the number of false measurements generated by the exteroceptive sensor mounted on each vehicle is Poisson distributed with an average of 5 and possesses a uniform spatial distribution over the sensor FOV. The probability of detection of a feature during the sub-mapping process is assumed to be constant and set to 0.93 and the probability of survival of a landmark from the current time step to the next is also assumed to be constant and set to 0.95. The number of particles used at each vehicle is 100. The control commands (heading and velocity) are updated at 40 Hz, and the observation update occurs at 5 Hz.

At the RFS HCMSLAM filter, it is assumed that the number of false landmarks present in the sub-map produced by each vehicle is Poisson distributed with an average of 1 and possesses a uniform spatial distribution over the entire sub-map.

The probability of detection of a landmark due the sub-mapping process is assumed to be constant and set to 0.95, and the probability of survival of a landmark in the global landmark map from the current time step to the next is assumed to be constant and set to 0.95. The measurement noise in terms of displacement in x and y directions (σ_x and σ_y) are set to 0.5m each. The number of particles used is 500 and the filter runs at 0.1 Hz.

Fig.5.4 and Fig.5.5 compare the estimated and the actual vehicle trajectories of a sample run of proposed HCMSLAM solution against the RFS-CMSLAM-II solution (see Chapter 3). It is clear that the proposed HCMSLAM solution produces comparable results to the centralized RFS CMSLAM solution. RMS vehicle position errors are compared in Fig.5.6. It is apparent that the estimated positional error values are acceptably smaller during the loop closure. Moreover it is clear that an improved positional accuracy can be obtained due to the overlapping regions of the vehicle trajectories. Average number of landmarks present in the map is compared against the actual number of observed landmarks in Fig.5.7. Average map error in terms of OSPA distance [77] is compared in Fig.5.8, it is clear that the proposed HCMSLAM solution tends to produce smaller map error over time. Average running times are compared in Fig.5.10 and Fig.5.9. It is clear that the simulation time taken by HCMSLAM solution to extract the global landmark map and estimate the global vehicle trajectories at each step are much more smaller than the time taken to produce a sub-map at 0.1 Hz. As a result, the proposed solution is much more robust against delays in communication and computational delays at the fusion node.

(a) RFS-CMSLAM-II

Figure 5.4: Estimated vehicle trajectories (vehicle 1 in green and vehicle 2 in red) along with estimated feature locations (in cyan circles).

(a) RFS-HCMSLAM

Figure 5.5: Estimated vehicle trajectories (vehicle 1 in green and vehicle 2 in red) along with estimated feature locations (in cyan circles).

(a) Vehicle 1

(b) Vehicle 2

Figure 5.6: RMS Vehicle position errors of 50 MC runs.

Figure 5.7: Average number of landmarks in the map, superimposed on the actual number of landmarks (in black).

Figure 5.8: Average map error in terms of OSPA distance.

Figure 5.9: Average simulation time taken to produce a sub-map (RFS mono-SLAM).

Figure 5.10: Average simulation time (seconds) taken to fuse sub-maps (RFS HCM-SLAM).

Table 5.1: Parameters Used in the Simulation

Vehicle Parameters		Values
Velocity	V	3m/s
Sensor FOV	Range (r)	0 - 35m
	Bearing (b)	$-\pi$ - $+\pi$
Control Noise	Velocity (σ_v)	0.2m/s
	Steering Angle (σ_a)	1^0
Measurement Noise	Range (σ_r)	0.1m
	Bearing (σ_b)	1^0

5.8 Conclusion

In this chapter we have presented a novel hierarchical solution to the Collaborative Multi-Vehicle SLAM (CMSLAM) problem by introducing a random finite set (RFS) mono-SLAM based sub-mapping process.

Instead of fusing the control and measurement information from all collaborating vehicles at each time step, we propose a RFS mono-SLAM based sub-mapping process where each collaborating vehicle produces a local sub-map by observing the features in the vicinity. The resultant sub-maps along with the sequences of applied control commands during the sub-mapping process are fused at a higher level hierarchical CMSLAM (HCMSLAM) algorithm with a much lower frequency, resulting reduced network bandwidth and computational requirements.

The HCMSLAM posterior is evaluated by factorizing into a product of the joint vehicle trajectories posterior and the global landmark map posterior conditioned on the vehicle trajectories. The global landmark map and the measurements are modeled as RFSs, yielding a Bayesian Hierarchical CMSLAM solution with inbuilt map management, data association and clutter filtering. Since the sub-mapping process at individual vehicle does not require any information/results back from the fusion node, proposed solution doesn't compromise the accuracy of the SLAM

process due to communication delays. Any communication delay could only affect other higher level processes that rely on the global landmark map at a collaborating vehicle. Moreover, it is also clear that the proposed approach doesn't suffer from the data incest problem since the local sub-map and the vehicle trajectory is reset in each vehicle once the sub-map and the control commands are communicated into the higher level fusion node. Performance of the proposed solution is evaluated using a simulation.

Chapter 6

Conclusion and Future Work

This Chapter summarizes the work presented in this thesis and discusses its major findings. Section 6.1 presents a chapter by chapter summary of the thesis and the contributions made into the robotics research. Section 6.2 presents the overall contribution of the thesis into the autonomous mobile robot navigation. Future research directions are summarized in section 6.3

6.1 Summary

In Chapter 1 we present the motivation, objectives and the contributions of the thesis. Use of autonomous multi-vehicle systems can greatly enhance performance gains in terms of speed, accuracy, reliability and robustness of complex missions involving larger areas. In order to effectively utilize such a system, it is required to localize the autonomous vehicles which are deployed in the possibly unknown and unstructured environment. In this thesis we propose to overcome this localization problem through Collaborative Multi-Vehicle Localization and Mapping (CMSLAM).

In Chapter 2 we present an overview of state of the art SLAM solutions. A brief

introduction to the SLAM problem is presented describing the importance of SLAM in accurate navigation of truly autonomous mobile robots. There is a plethora of Single-Vehicle SLAM solutions that exist in mobile robotics literature, out of which extended Kalman filter based SLAM (EKF-SLAM), FastSLAM and RFS SLAM can be considered as most influential algorithms to date. Both EKF-SLAM and FastSLAM solutions are conventional vector based SLAM solutions. As a result, in addition to the actual SLAM filter, it is required to solve several additional sub-problems such as data association, clutter filtering and landmark map management outside the Bayesian SLAM recursion in order to produce a consistent SLAM solution. Assumption of clutter free measurements and ideal landmark detection and data association limit the practical application of such solutions. It can be concluded that all existing Multi-Vehicle SLAM solutions stem from these solutions inherit above mentioned issues.

In Chapter 3 a novel solution to the collaborative multi-vehicle SLAM (CMSLAM) problem is presented, by extending the recently developed mono-SLAM filter framework based on random finite set (RFS) representation and finite set statistical (FISST) methods. By appropriately modeling the landmark map and measurements as RFSs and formulating the CMSLAM problem by adopting FISST techniques, two new multi-sensor information fusion techniques for SLAM were introduced, yielding two Bayesian CMSLAM algorithms with inbuilt map management, data association and clutter filtering, while catering for landmark detection uncertainty and data association uncertainty. Furthermore, the particle filter based joint vehicle trajectory posterior estimation method enables adopting complex motion models, catering for various adverse field conditions that affect vehicular motion in challenging environments. It is evident from the simulation results and actual experimental results that the proposed RFS-CMSLAM algorithms perform effectively under measurement clutter condition with data association uncertainty.

In Chapter 4 a novel solution to the Collaborative Multi-Vehicle Simultaneous Localization and Mapping with Moving Object Tracking (CMSLAMMOT) problem is presented by extending our previous work on random finite set based CMSLAM solution presented in chapter 3. We have appropriately modeled the measurements and features (both static and dynamic) as RFSs and formulated a Bayesian solution, where both static and dynamic features are tracked from measurements with clutter and come up with a robust CMSLAMMOT algorithm. The proposed solution inherits inbuilt map management, data association and clutter filtering, while accommodating for feature detection, feature survival uncertainty and data association uncertainty. The joint vehicle trajectory posterior is estimated using a particle filter, which enables adopting complex non-linear motion models. It is evident from the simulation results and actual experiment results the proposed solution performs effectively under measurement clutter conditions and data association uncertainty. Furthermore the proposed CMSLAMMOT algorithm produces comparable results to the proposed RFS CMSLAM solution (Chapter 3) in terms of positional error while capably extracting moving objects present in the sensor FOV.

In Chapter 5 we have presented a novel hierarchical solution to the Collaborative Multi-Vehicle SLAM (CMSLAM) problem by introducing a random finite set Single-Vehicle SLAM based sub-mapping process. Instead of fusing the control and measurement information at each time step, the resultant sub-maps along with the sequences of applied control commands recorded during the sub-mapping process are fused at a higher level hierarchical CMSLAM algorithm (HCMSLAM) with a much lower frequency, resulting reduced network bandwidth and less computational power requirements at both fusion node and individual vehicle. The HCMSLAM posterior is evaluated by factorizing into a product of the joint vehicle trajectories posterior and the landmark map posterior conditioned on the vehicle trajectories. The global landmark map and the measurements are modeled as RFSs, yielding a

Bayesian Hierarchical CMSLAM solution with inbuilt map management, data association and clutter filtering while catering for landmark detection uncertainty and data association uncertainty. Performance of the proposed HCMSLAM solution is demonstrated via simulation results.

6.2 Conclusion

In this work we have addressed some of the key limitations of existing Collaborative Multi-Vehicle SLAM (CMSLAM) solutions. Almost all state of the art CMSLAM algorithms stem from conventional vector based SLAM solutions such as EKF-SLAM and FastSLAM or their variants. As a result, in addition to the actual SLAM filter, it is required to solve several additional sub-problems such as data association, clutter filtering and landmark map management outside the Bayesian SLAM recursion in order to produce a consistent CMSLAM solution. Moreover, assumption of clutter free measurements and ideal landmark detection and data association limit the practical application of such solutions. These limitations have to be properly addressed in order to practically apply such CMSLAM solutions.

By appropriately modeling the landmark map and measurements as RFSs and formulating the CMSLAM problem by adopting FISST techniques, we have developed three variants of Bayesian CMSLAM algorithms based on recently developed multi-sensor information fusion methods in FISST. All proposed solutions yield inbuilt map management, data association and clutter filtering, while catering for landmark detection uncertainty and data association uncertainty. In a true autonomous Multi-Vehicle setup, detection and tracking of moving objects present in the sensor FOV has to be considered as a crucial navigation safety requirement. Moreover, this broadens the range of applicable solutions from various civilian applications

to complex military solutions involving larger number of autonomous vehicles. In order to address this requirement, we proposed the CMSLAMMOT solution where, in addition to solving the CMSLAM problem, detection and tracking of moving objects (DATMO) is also performed. Although centralized solutions cater statistically principled data fusion methods, in practice, such solutions limit the practical applicability of CMSLAM solutions as the number of collaborating vehicles increase. This is mainly due to the exponentially increasing communication bandwidth and the computational requirements in the fusion node. In order to address this issue we have proposed the RFS HCMSLAM solution, where instead of fusing control and observation information from all collaborating vehicles at each time step, a RFS single-vehicle SLAM based sub-mapping process is introduced. It is evident from the simulation and experimental results that all the proposed algorithms perform effectively under measurement clutter and detection uncertainty, yielding consistent estimates of the feature locations and adequately small positional errors.

6.3 Future research directions

The work presented in this thesis can be considered as the basis for developing the navigation platform of a large array of practical autonomous Multi-Vehicle applications. Thus far we have been able to solve some of the key pre-requisites on our way towards truly autonomous Multi-Vehicle systems. These solutions can be further improved upon identifying practical limitations by applying them in dynamic unknown and unstructured environments such as those found in marine environments. Some of the possible research directions that we have identified for further investigation are as follows.

- The proposed solutions assumes that the motion of the vehicles occur in a 2D

plane and the measurements are also acquired by sensing the environment on a 2D plane. But for practical CMSLAM solutions, it would be useful if the proposed solutions are extended to cater for 3D case, which we may encounter in complex environment such as those found in marine environments.

- The cardinalized PHD filter (CPHD) [55] was recently proposed as a better multi-target tracking solution than the PHD filter for high clutter conditions. In the future implementations, a CPHD filter can be used instead of the PHD filter in order to improve the estimation performance of features under high clutter conditions.
- In the proposed solutions, we assume that the clutter rate is known, however, in practice this may raise issues if the expected clutter rate is too smaller or too higher than the expected clutter rate. A novel CPHD filter was developed by Mahler et al. in [105], which address this issue and derives a CPHD filter capable of multi-target tracking with unknown clutter rate. Hence future developements could benefit from using such a CPHD filter in place of the proposed PHD filter based solutions for practical applications.
- Although the proposed solutions are mathematically rigorous, one of the major limitations of the current solutions is that, in practice a large number of vehicles can produce inferior results due to various complications arise during data fusion as a result of the communication delays and computational load at the fusion node.

Hence, a fully distributed CMSLAM approach with delayed information fusion is highly desirable for truly autonomous Multi-Vehicle applications involving a large number of vehicles. Such a distributed system can cater for local information fusion, meaning nearby vehicles could share information among them and produce local results and such results can be shared among each

other vehicles as they are available in the communication range. Such a system could obtain the same results as a centralized CMSLAM solution, while eliminating the necessity for expensive long-range communication links and powerful fusion nodes. We hope our work provides the basis for developing such algorithms in future.

Author's Publications

Journal Papers

- Diluka Moratuwage, Danwei Wang, Akshay Rao, Namal Senarathna and Han Wang. RFS Collaborative Multi-Vehicle SLAM in Dynamic High Clutter Environments. Robotics and Automation Magazine

Conference Papers

- D. Moratuwage, B.N. Vo and D. Wang. Collaborative Multi-Vehicle SLAM with Moving Object Tracking. IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, May, 2013
- D. Moratuwage, B.N. Vo and D. Wang. A Hierarchical approach to the Multi-Vehicle SLAM problem. Information Fusion (FUSION), Singapore, July, 2012
- D. Moratuwage, B.N. Vo, D. Wang and H. Wang. Extending Bayesian RFS SLAM to Multi-Vehicle SLAM. IEEE Control Automation Robotics & Vision (ICARCV), Guangzhou, China, 2010
- Moratuwage, M.D.P., Wijesoma W.S., Kalyan, B., Dong J.F., Namal Senarathne P.G.C., Hover F.S., Patrikalakis N.M. Collaborative multi-vehicle localization

and mapping in marine environments. IEEE OCEANS, Sydney, Australia, May, 2010

- Moratuwage M.D.P., Wijesoma W.S., Kalyan B., Patrikalakis N.M., Moghadam P. Collaborative multi-vehicle localization and mapping in high clutter environments. IEEE Control Automation Robotics & Vision (ICARCV), Singapore, December, 2010
- Kalyan B., Lee K.W., Wijesoma S., Moratuwage D., Patrikalakis N.M. A random finite set based detection and tracking using 3D LIDAR in dynamic environments. IEEE Systems Man and Cybernetics (SMC), Istanbul, Turkey, October, 2010

Appendix A

GM-PHD filter in RFS CMSLAM

In this chapter, we outline the Gaussian mixture implementation of the PHD filters ("General multi-sensor PHD update method" and the "Iterative multi-sensor PHD update method") presented in the proposed RFS CMSLAM solutions (Chapter 3). It is assumed that the PHD of the landmark map at time $k - 1$ is represented as a mixture of Gaussians, and the newly appearing landmarks in the joint sensor FOV at time k can be represented as a mixture of Gaussians, then the resultant measurement updated PHD also results in a mixture of Gaussians.

Let the PHD of the map posterior at time $k - 1$, correspond to i th particle be denoted by a Gaussian mixture of the form,

$$D_{k-1|k-1}(\zeta_{k-1} | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k-1}^{(1),[i]}, X_{0:k-1}^{(2),[i]}) = \sum_{j=1}^{J_{k-1}} \eta_{k-1}^{[i,j]} \mathcal{N}(\zeta_{k-1}; \mu_{k-1}^{[i,j]}, P_{k-1}^{[i,j]}) \quad (\text{A.1})$$

where, J_{k-1} is the number of Gaussians, while $\eta_{k-1}^{[i,j]}$, $\mu_{k-1}^{[i,j]}$ and $P_{k-1}^{[i,j]}$ respectively denote weight, mean and covariance of the j^{th} Gaussian. Intensity of the new landmarks appearing in the joint sensor FOV of vehicles at time k can also be

represented as a mixture of Gaussians as follows,

$$b_k(\zeta_k | X_k^{(1),[i]}, X_k^{(2),[i]}) = \sum_{j=1}^{J_{b,k}} \eta_{b,k}^{[i,j]} \mathcal{N}(\zeta_k; \mu_{b,k}^{[i,j]}, P_{b,k}^{[i,j]}) \quad (\text{A.2})$$

where, $J_{b,k}$ denotes the expected number of new landmarks appearing in the joint FOV of the two vehicles. Therefore, PHD of the predicted landmark map posterior (3.17) simplifies into a mixture of Gaussians of the form,

$$D_{k|k-1}(\zeta_k | Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}, X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}) = \sum_{j=1}^{J_{k|k-1}} \eta_{k|k-1}^{[i,j]} \mathcal{N}(\zeta_{k|k-1}; \mu_{k|k-1}^{[i,j]}, P_{k|k-1}^{[i,j]}) \quad (\text{A.3})$$

Assume that the measurement model is given by a non-linear function of the form,

$$z_k^{(r)} = h_k^{(r)}(\zeta_k, X_k^{(r)}, \xi_k^{(r)}) \quad (\text{A.4})$$

where $\xi_k^{(r)}$ is the zero-mean Gaussian noise with covariance $R_k^{(r)}$. We follow the extended Kalman filter (EKF) based measurement update method for the GM-PHD filter [51] by linearizing the measurement model and approximating the measurement likelihood function as follows,

$$g_k^{(r)}(z^{(r)} | \zeta_k, X_k^{(r)}) \approx \mathcal{N}(z^{(r)}; h_k^{(r)}(\zeta_k, X_k^{(r)}, 0), U_k R_k U_k^T) \quad (\text{A.5})$$

where U_k is the Jacobian of $h_k^{(r)}(\zeta_k, X_k^{(r)}, \xi_k)$ with respect to ξ_k at $\xi_k = 0$. Let, $H_k^{(r)}$ denote the Jacobian of $h_k^{(r)}(\zeta_k, X_k^{(r)}, 0)$ with respect to ζ_k at some $\zeta_k = \mu$. Then we can show that the measurement updated PHD of the map posterior results in a mixture of Gaussians in both CMSLAM solutions proposed in Chapter 3, as shown in Table A.1 and Table A.5.

Table A.1: PHD Filter Update - Iterative method

```

1. Given:  $D_{k|k-1}^{[i]}(\zeta_k | X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}, Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}) = \{\eta_{k|k-1}^{[i,j]}, \mu_{k|k-1}^{[i,j]}, P_{k|k-1}^{[i,j]}\}_{j=1}^{J_{k|k-1}^{[i]}}$ 
2. for  $i = 1$  to  $N_s$ 
3.   prepare_for_itr_update
4.   for  $j = 1$  to  $J_{k|k-1}^{[i]}$  (Missed detections - vehicle 1)
5.      $\eta_k^{[i,j]} = (1 - P_D^{(1)})\eta_{k|k-1}^{[i,j]}$ 
6.      $\mu_k^{[i,j]} = \mu_{k|k-1}^{[i,j]}$ 
7.      $P_k^{[i,j]} = P_{k|k-1}^{[i,j]}$ 
8.   end
9.    $l = 0$ 
10.  for each  $z_k^{(1)} \in Z_k^{(1)}$  (Detections)
11.    itr_update_ $z_k^{(1)}$ 
12.  end
13.   $J_{k|k}^{(1),[i]} = (l + 1)J_{k|k-1}^{[i]}$ 
14.   $l = 0$ 
15.  for  $j = 1$  to  $J_k^{(1),[i]}$  (Missed detections - vehicle 2)
16.     $\eta_k^{[i,j]} = (1 - P_D^{(2)})\eta_k^{[i,j]}$ 
17.     $\mu_k^{[i,j]} = \mu_k^{[i,j]}$ 
18.     $P_k^{[i,j]} = P_k^{[i,j]}$ 
19.  end
20.  for each  $z_k^{(2)} \in Z_k^{(2)}$  (Detections - vehicle 2)
21.    itr_update_ $z_k^{(2)}$ 
22.  end
23.   $J_{k|k}^{[i]} = (l + 1)J_{k|k}^{(1),[i]}$ 
24. end
25. Output:  $D_k^{[i]}(\zeta_k | X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}, Z_{1:k}^{(1)}, Z_{1:k}^{(2)}) = \{\eta_k^{[i,j]}, \mu_k^{[i,j]}, P_k^{[i,j]}\}_{j=1}^{J_{k|k}^{[i]}}$ 

```

Table A.2: prepare_for_itr_update

```

1.   for  $j = 1$  to  $J_{k|k-1}^{[i]}$  (Preparing for update)
2.      $S_k^{(1),[i,j]} = U_k^{(1),[i,j]} R_k^{(1),[i,j]} U_k^{(1),[i,j]T} + H_k^{(1),[i,j]} P_{k|k-1}^{[i,j]} H_k^{(1),[i,j]T}$ 
3.      $K_k^{(1),[i,j]} = P_{k|k-1}^{[i,j]} H_k^{(1),[i,j]} (S_k^{(1),[i,j]})^{-1}$ 
4.      $q_k^{(1),[i,j]}(z_k^{(1)}) = \mathcal{N}(z_k^{(1)}; h_k^{(1)}(\mu_{k|k-1}^{[i,j]}, X_k^{(1),[i]}, 0), S_k^{(1),[i,j]})$ 
5.   end

```

Table A.3: itr_update_ $z_k^{(1)}$

```

1.    $l = l + 1$ 
2.   for  $j = 1$  to  $J_{k|k-1}^{[i]}$ 
3.      $\tau^{(1),[j]} = P_D^{(1)} \eta_k^{[i,j]} q_k^{(1),[i,j]}(z_k^{(1)})$ 
4.      $\mu_k^{(1),[i,lJ_k^{[i]}+j]} = \mu_{k|k-1}^{[i,j]} + K_k^{(1),[i,j]}(z_k^{(1)} - h_k^{(1)}(\mu_{k|k-1}^{[i,j]}, X_k^{(1),[i]}, 0))$ 
5.      $P_k^{(1),[i,lJ_k^{[i]}+j]} = (I - K_k^{(1),[i,j]} H_k^{(1),[i,j]}) P_{k|k-1}^{[i,j]}$ 
6.   end
7.   for  $j = 1$  to  $J_{k|k-1}^{[i]}$ 
8.      $\eta_k^{(1),[i,lJ_k^{[i]}+j]} = \tau^{(1),[j]} / c^{(1)}(z_k^{(1)}) + \sum_{m=1}^{J_{k|k-1}^{[i]}} \tau^{(1),[m]}$ 
9.   end

```

Table A.4: itr_update_ $z_k^{(2)}$

```

1.    $l = l + 1$ 
2.   for  $j = 1$  to  $J_{k|k}^{(1),[i]}$ 
3.      $S_k^{(1,2),[i,j]} = U_k^{(2),[i,j]} R_k^{(2),[i,j]} U_k^{(2),[i,j]T} + H_k^{(2),[i,j]} P_{k|k}^{(1),[i,j]} H_k^{(2),[i,j]T}$ 
4.      $K_k^{(1,2),[i,j]} = P_{k|k}^{(1),[i,j]} H_k^{(2),[i,j]} (S_k^{(1,2),[i,j]})^{-1}$ 
5.      $q_k^{(1,2),[i,j]}(z_k^{(2)}) = \mathcal{N}(z_k^{(2)}; h_k^{(2)}(\mu_{k|k}^{(1),[i,j]}, X_k^{(2),[i]}, 0), S_k^{(1,2),[i,j]})$ 
6.      $\tau^{[j]} = P_D^{(2)} \eta_k^{(1),[i,j]} q_k^{(1,2),[i,j]}(z_k^{(2)})$ 
7.      $\mu_k^{[i,lJ_k^{[i]}+j]} = \mu_{k|k}^{(1),[i,j]} + K_k^{(1,2),[i,j]}(z_k^{(2)} - h_k^{(2)}(\mu_{k|k}^{(1),[i,j]}, X_k^{(2),[i]}, 0))$ 
8.      $P_k^{[i,lJ_k^{[i]}+j]} = (I - K_k^{(1,2),[i,j]} H_k^{(2),[i,j]}) P_{k|k}^{(1),[i,j]}$ 
9.   end
10.  for  $j = 1$  to  $J_k^{(1),[i]}$ 
11.     $\eta_k^{[i,lJ_k^{[i]}+j]} = \tau^{[j]} / c^{(2)}(z_k^{(2)}) + \sum_{m=1}^{J_{k|k}^{[i]}} \tau^{[m]}$ 
12.  end

```

Table A.5: PHD Filter update - General multi-sensor method

```

1. Given:  $D_{k|k-1}^{[i]}(\zeta_k | X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}, Z_{1:k-1}^{(1)}, Z_{1:k-1}^{(2)}) = \{\eta_{k|k-1}^{[i,j]}, \mu_{k|k-1}^{[i,j]}, P_{k|k-1}^{[i,j]}\}_{j=1}^{J_{k|k-1}^{[i]}}$ 
2. for  $i = 1$  to  $N_s$ 
3.   prepare_for_gen_update
4.    $l = 0$ 
5.   for each partition  $\mathcal{P}_m, m = 1, 2, \dots, |\mathcal{P}_k|$ 
6.      $d_W^{(m)} = 1$ 
7.      $l_{part\_start} = l$ 
8.     for each  $W \in \mathcal{P}_m$ 
9.       if  $W = \{z_k^{(1)}\}$ 
10.        gen_update_for_ $\{z_k^{(1)}\}$ 
11.      end
12.      if  $W = \{z_k^{(2)}\}$ 
13.        gen_update_for_ $\{z_k^{(2)}\}$ 
14.      end
15.      if  $W = \{z_k^{(1)}, z_k^{(2)}\}$ 
16.        gen_update_for_ $\{z_k^{(1)}, z_k^{(2)}\}$ 
17.      end
18.    end
19.     $l_{part\_end} = l$ 
20.    for  $l_{part} = l_{part\_start}$  to  $l_{part\_end}$ 
21.      for  $j = 1$  to  $J_{k|k-1}^{[i]}$ 
22.         $\eta_k^{[i, l_{part} J_{k|k-1}^{[i]} + j]} = \eta_k^{[i, l_{part} J_{k|k-1}^{[i]} + j]} * d_W^{(m)}$ 
23.      end
24.    end
25.  end
26.   $l_{part\_end} = l$ 
27.  for  $l_{part} = l_{part\_start}$  to  $l_{part\_end}$ 
28.    for  $j = 1$  to  $J_{k|k-1}^{[i]}$ 
29.       $\eta_k^{[i, l_{part} J_{k|k-1}^{[i]} + j]} = \eta_k^{[i, l_{part} J_{k|k-1}^{[i]} + j]} / (\sum_{m=1}^{|\mathcal{P}_k|} d_W^{(m)})$ 
30.    end
31.  end
32.   $J_{k|k}^{[i]} = (l + 1) J_{k|k-1}^{[i]}$ 
33. Output:  $D_{k|k}^{[i]}(\zeta_k | X_{0:k}^{(1),[i]}, X_{0:k}^{(2),[i]}, Z_{1:k}^{(1)}, Z_{1:k}^{(2)}) = \{\eta_k^{[i,j]}, \mu_k^{[i,j]}, P_k^{[i,j]}\}_{j=1}^{J_k^{[i]}}$ 

```

Table A.6: prepare_for_gen_update

-
1. for $j = 1$ to $J_{k|k-1}^{[i]}$ (Preparing for update)
 2. $S_k^{(1),[i,j]} = U_k^{(1),[i,j]} R_k^{(1),[i,j]} U_k^{(1),[i,j]T} + H_k^{(1),[i,j]} P_{k|k-1}^{[i,j]} H_k^{(1),[i,j]T}$
 3. $S_k^{(2),[i,j]} = U_k^{(2),[i,j]} R_k^{(2),[i,j]} U_k^{(2),[i,j]T} + H_k^{(2),[i,j]} P_{k|k-1}^{[i,j]} H_k^{(2),[i,j]T}$
 4. $S_k^{(1,2),[i,j]} = U_k^{(2),[i,j]} R_k^{(2),[i,j]} U_k^{(2),[i,j]T} + H_k^{(2),[i,j]} P_{k|k-1}^{(1),[i,j]} H_k^{(2),[i,j]T}$
 5. $K_k^{(1),[i,j]} = P_{k|k-1}^{[i,j]} H_k^{(1),[i,j]T} (S_k^{(1),[i,j]})^{-1}$
 6. $K_k^{(2),[i,j]} = P_{k|k-1}^{[i,j]} H_k^{(2),[i,j]T} (S_k^{(2),[i,j]})^{-1}$
 7. $K_k^{(1,2),[i,j]} = P_{k|k-1}^{(1),[i,j]} H_k^{(2),[i,j]T} (S_k^{(2),[i,j]})^{-1}$
 8. $q_k^{(1),[i,j]}(z_k^{(1)}) = \mathcal{N}(z_k^{(1)}; h_k^{(1)}(\mu_{k|k-1}^{[i,j]}, X_k^{(1),[i]}, 0), S_k^{(1),[i,j]})$
 9. $q_k^{(2),[i,j]}(z_k^{(2)}) = \mathcal{N}(z_k^{(2)}; h_k^{(2)}(\mu_{k|k-1}^{[i,j]}, X_k^{(2),[i]}, 0), S_k^{(2),[i,j]})$
 10. $q_k^{(1,2),[i,j]}(z_k^{(2)}) = \mathcal{N}(z_k^{(2)}; h_k^{(2)}(\mu_{k|k-1}^{(1),[i,j]}, X_k^{(2),[i]}, 0), S_k^{(1,2),[i,j]})$
 11. end
 12. for $j = 1$ to $J_{k|k-1}^{[i]}$ (Missed detections - vehicle 1 & vehicle 2)
 13. $\eta_k^{[i,j]} = (1 - P_D^{(1)})(1 - P_D^{(2)})\eta_{k|k-1}^{[i,j]}$
 14. $\mu_k^{[i,j]} = \mu_{k|k-1}^{[i,j]}$
 15. $P_k^{[i,j]} = P_{k|k-1}^{[i,j]}$
 16. end

Table A.7: gen_update_for_ $\{z_k^{(1)}\}$

-
1. $l = l + 1$
 2. for $j = 1$ to $J_{k|k-1}^{[i]}$
 3. $\tau^{[j]} = P_D^{(1)}(1 - P_D^{(2)})\eta_{k|k-1}^{[i,j]} q_k^{(1),[i,j]}(z_k^{(1)})$
 4. $\mu_k^{[i,lJ_{k|k-1}^{[i]}+j]} = \mu_{k|k-1}^{[i,j]} + K_k^{(1),[i,j]}(z_k^{(1)} - h_k^{(1)}(\mu_{k|k-1}^{[i,j]}, X_k^{(1),[i]}, 0))$
 5. $P_k^{[i,lJ_{k|k-1}^{[i]}+j]} = (I - K_k^{(1),[i,j]} H_k^{(1),[i,j]}) P_{k|k-1}^{[i,j]}$
 6. end
 7. for $j = 1$ to $J_{k|k-1}^{[i]}$
 8. $\eta_k^{[i,lJ_{k|k-1}^{[i]}+j]} = \tau^{[j]} / \lambda_c^{(1)} c_k^{(1)}(z_k^{(1)}) + \sum_{m=1}^{J_{k|k-1}^{[i]}} \tau^{[m]}$
 9. end
 10. $d_W^{(m)} = d_W^{(m)} * ((\lambda_c^{(1)} c_k^{(1)}(z_k^{(1)}) + \sum_{m=1}^{J_{k|k-1}^{[i]}} \tau^{[m]}) / \lambda_c^{(1)} c_k^{(1)}(z_k^{(1)}))$

Table A.8: gen_update_for_{\{z_k^{(2)}\}}

-
1. $l = l + 1$
 2. for $j = 1$ to $J_{k|k-1}^{[i]}$
 3. $\tau^{[j]} = P_D^{(2)}(1 - P_D^{(1)})\eta_{k|k-1}^{[i,j]}q_k^{(2),[i,j]}(z_k^{(2)})$
 4. $\mu_k^{[i,lJ_{k|k-1}^{[i]}+j]} = \mu_{k|k-1}^{[i,j]} + K_k^{(2),[i,j]}(z_k^{(2)} - h_k^{(2)}(\mu_{k|k-1}^{[i,j]}, X_k^{(1),[i]}, 0))$
 5. $P_k^{[i,lJ_{k|k-1}^{[i]}+j]} = (I - K_k^{(2),[i,j]}H_k^{(2),[i,j]})P_{k|k-1}^{[i,j]}$
 6. end
 7. for $j = 1$ to $J_{k|k-1}^{[i]}$
 8. $\eta_k^{[i,lJ_{k|k-1}^{[i]}+j]} = \tau^{[j]}/\lambda_c^{(2)}c_k^{(2)}(z_k^{(2)}) + \sum_{m=1}^{J_{k|k-1}^{[i]}} \tau^{[m]}$
 9. end
 - 10.

Table A.9: gen_update_for_{\{z_k^{(1)}, z_k^{(2)}\}}

-
1. $l = l + 1$
 2. for $j = 1$ to $J_{k|k-1}^{[i]}$
 3. $\mu_k^{(1),[i,j]} = \mu_{k|k-1}^{[i,j]} + K_k^{(1),[i,j]}(z_k^{(1)} - h_k^{(1)}(\mu_{k|k-1}^{[i,j]}, X_k^{(1),[i]}, 0))$
 4. $P_k^{(1),[i,j]} = (I - K_k^{(1),[i,j]}H_k^{(1),[i,j]})P_{k|k-1}^{[i,j]}$
 5. $S_k^{(1,2),[i,j]} = U_k^{(2),[i,j]}R_k^{(2),[i,j]}U_k^{(2),[i,j]T} + H_k^{(2),[i,j]}P_{k|k}^{(1),[i,j]}H_k^{(2),[i,j]T}$
 6. $K_k^{(1,2),[i,j]} = P_{k|k}^{(1),[i,j]}H_k^{(2),[i,j]T}(S_k^{(2),[i,j]})^{-1}$
 7. $K_k^{(1,2),[i,j]} = P_{k|k}^{(1),[i,j]}H_k^{(2),[i,j]T}(S_k^{(2),[i,j]})^{-1}$
 8. $q_k^{(1,2),[i,j]}(z_k^{(2)}) = \mathcal{N}(z_k^{(2)}; h_k^{(2)}(\mu_{k|k}^{(1),[i,j]}, X_k^{(2),[i]}, 0), S_k^{(1,2),[i,j]})$
 9. $\tau^{[j]} = P_D^{(1)}P_D^{(2)}\eta_{k|k-1}^{[i,j]}q_k^{(1),[i,j]}(z_k^{(1)})q_k^{(1,2),[i,j]}(z_k^{(2)})$
 10. $\mu_{k|k}^{[i,lJ_{k|k-1}^{[i]}+j]} = \mu_{k|k-1}^{(1),[i,j]} + K_k^{(2),[i,j]}(z_k^{(2)} - h_k^{(2)}(\mu_{k|k-1}^{(1),[i,j]}, X_k^{(2),[i]}, 0))$
 11. $P_{k|k}^{[i,lJ_{k|k-1}^{[i]}+j]} = (I - K_k^{(2),[i,j]}H_k^{(2),[i,j]})P_{k|k}^{(1),[i,j]}$
 12. end
 13. for $j = 1$ to $J_{k|k-1}^{[i]}$
 14. $\eta_k^{[i,lJ_{k|k-1}^{[i]}+j]} = \tau^{[j]}/\sum_{m=1}^{J_{k|k-1}^{[i]}} \tau^{[m]}$
 15. end
 16. $d_W^{(m)} = d_W^{(m)} * ((\sum_{m=1}^{J_{k|k-1}^{[i]}} \tau^{[m]})/\lambda_c^{(1)}c_k^{(1)}(z_k^{(1)})\lambda_c^{(2)}c_k^{(2)}(z_k^{(2)}))$

Bibliography

- [1] G. Dissanayake, J. Paxman, J.V. Miro, O. Thane and H.T. Thi, “Robotics for Urban Search and Rescue,” in *IEEE Int. Conf. Industrial and Inf. Sys.*, August 2006.
- [2] A. Davids, “Urban Search and Rescue Robots: From Tragedy to Technology,” *IEEE Intel. Sys.*, vol. 17, no. 2, pp. 81–83, 2002.
- [3] R.R. Murphy, J. Kravitz, K. Peligren, J. Milward and J. Stanway, “Preliminary Report: Rescue Robot at Crandall Canyon, Utah, Mine Disaster,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008.
- [4] R.W. Beard, T.W. McLain, D.B. Nelson and D. Kingston, “Decentralized Cooperative Aerial Surveillance Using Fixed-Wing Miniature UAVs,” *Proc. of the IEEE*, vol. 94, no. 7, pp. 1306–1324, 2006.
- [5] H. Johannsson, M. Kaess, B. Englot, F. Hover and J. Leonard, “Imaging Sonar-Aided Navigation for Autonomous Underwater Harbor Surveillance,” in *Proc. IEEE Int. Conf. Intel. Robot. Sys.*, October 2010.
- [6] J.E. Manley, “Multiple AUV missions in the National Oceanic and Atmospheric Administration,” in *IEEE/OES Autonomous Underwater Vehicles*, 2004.
- [7] N.A. Tsokas and K.J. Kyriakopoulos, “Multi-Robot Multiple Hypothesis Tracking for Pedestrian Tracking with Detection Uncertainty,” in *Mediterranean Conf. Con. and Autom.*, June 2011.
- [8] M. Teutsch and W. Kruger, “Detection, Segmentation, and Tracking of Moving Objects in UAV Videos,” in *IEEE Int. Conf. Adv. Video and Signal-based Surveillance*, September 2012.
- [9] S. Scheding, G. Dissanayake, E.M. Nebot and H. Durrant-Whyte, “An experiment in autonomous navigation of an underground mining vehicle,” *IEEE Trans. Robot.*, vol. 15, no. 1, pp. 85–95, February 1999.
- [10] J.E. Manley, “Unmanned surface vehicles, 15 years of development,” in *IEEE OCEANS*, September 2008.

- [11] D. Spenneberg, C. Waldmann and R. Babb, "Exploration of Underwater Structures with Cooperative Heterogeneous Robots," in *IEEE OCEANS*, 2005.
- [12] A.J. Shafer, M.R. Benjamin, J.J. Leonard and J. Curcio, "Autonomous Cooperation of Heterogeneous Platforms for Sea-Based Search Tasks," in *Proc. IEEE OCEANS*, 2006.
- [13] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, June 2006.
- [14] T. Bailey and H. Durrant-Whyte, "Simultaneous Localization and Mapping: part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sept 2006.
- [15] R. Henthorn, D.W. Caress, H. Thomas, R. McEwen, W.J. Kirkwood, C.K. Paull and R. Keaten, "High-Resolution Multibeam and Subbottom Surveys of Submarine Canyons, Deep-Sea Fan Channels, and Gas Seeps Using the MBARI Mapping AUV," in *IEEE OCEANS*, 2006.
- [16] R.C. Smith and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty," *The Int. J. Robot. Res.*, vol. 5, no. 4, pp. 56–68, December 1986.
- [17] R. Smith, M. Self, and R. Cheeseman, *Estimating uncertain spatial relationships in robotics*. New York, NY, USA: Springer-Verlag., 1990.
- [18] G. Dissanayake, P.M. Newman, S. Clark, H. Durrant-Whyte and M. Csorba, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 229–241, June 2001.
- [19] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. : The MIT Press, 2005.
- [20] M. Montemerlo, "FastSLAM: a factored solution to the simultaneous localization and mapping problem with unknown data association," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, 2003.
- [21] S. Huang and G. Dissanayake, "Convergence and Consistency Analysis for Extended Kalman Filter Based SLAM," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 1036–1049, October 2007.
- [22] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani and H. Durrant-Whyte, "Simultaneous Localization and Mapping with Sparse Extended Information Filters," *The Int. J. Robot. Res.*, vol. 23, no. 7-8, pp. 693–716, August 2004.
- [23] S.B. Williams, "Efficient Solutions to Autonomous Mapping and Navigation Problems," Ph.D. dissertation, University of Sydney, Australian Center for Field Robotics, 2001.

- [24] S.B. Williams, G. Dissanayake and H. Durrant-Whyte, "An efficient approach to the simultaneous localization and mapping problem," in *Proc. IEEE Int. Conf. Robot. Autom.*, August 2002, pp. 406–411.
- [25] P. Pinies and J.D. Tardos, "Large-Scale SLAM Building Conditionally Independent Local Maps: Application to Monocular Vision," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1094–1106, 2008.
- [26] P. Pinies, M. Lina, J.D. and Tardos, "CI-Graph: An efficient approach for large scale SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3913–3920.
- [27] R. Martinez-Cantin and J.A. Castellanos, "Unscented SLAM for large-scale outdoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, August 2005, pp. 3427–3432.
- [28] C. Estrada, J. Neira and J.D. Tardos, "Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments," *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 588–596, August 2005.
- [29] Z. Wang, S. Huang and G. Dissanayake, "D-SLAM: A Decoupled Solution to Simultaneous Localization and Mapping," *The Int. J. Robot. Res.*, vol. 26, no. 2, pp. 187–204, February 2007.
- [30] F. Dellaert, J. Carlson, V. Ila, K. Ni and C.E. Thorpe, "Subgraph-preconditioned conjugate gradients for large scale SLAM," in *Proc. IEEE Int. Conf. Intell. Robot. Sys.*, October 2010.
- [31] K. Konolige, "Large-scale map-making," in *AAAI'04 Proceedings of the 19th national conference on Artificial intelligence*, 2004, pp. 457–463.
- [32] J.F. Dong, W.S. Wijesoma and A.P. Shacklock, "An Efficient Rao-Blackwellized Genetic Algorithmic Filter for SLAM," in *Proc. IEEE Int. Robot. Autom.*, April 2007, pp. 2427–2432.
- [33] J.F. Dong, S. Wijesoma and A.P. Shacklock, "Extended rao-blackwellised genetic algorithmic filter SLAM in dynamic environment with raw sensor measurement," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Nov 2007, pp. 1473–1478.
- [34] D.J. Feng and W.S. Wijesoma, "Improving Rao-Blackwellised Genetic Algorithmic Filter SLAM Through Genetic Learning," in *Proc. IEEE Int. Conf. Control, Autom., Robot. Comp. Vision.*, December 2008.
- [35] L.M Paz, P. Jensfelt, J.D. Tardos, J.D. and J. Neira, "EKF SLAM updates in $O(n)$ with Divide and Conquer SLAM," in *Proc. IEEE Int. Robot. Autom.*, April 2007, pp. 1657–1663.

- [36] K. Ni, D. Steedly and F. Dellaert, "Tectonic SAM: Exact; outof-core; submap-based SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, April 2007.
- [37] M. Kaess, A. Ranganathan and F. Dellaert, "iSAM: Incremental Smoothing and Mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, December 2008.
- [38] H. Zhao, J. Sha, Y. Zhao, J. Xi, J. Cui, H. Zha and R. Shibasaki, "Detection and Tracking of Moving Objects at Intersections Using a Network of Laser Scanners," *IEEE Trans. Intel. Transport. Sys.*, vol. 13, no. 2, pp. 655–670, June 2012.
- [39] C.C. Wang and C. Thorpe, "Online Simultaneous Localization And Mapping with Detection And Tracking of Moving Objects: Theory and Results from a Ground Vehicle in Crowded Urban Areas," in *Proc. IEEE Int. Conf. Robot. Autom.*, September 2003, pp. 842–849.
- [40] C.C. Wang, C. Thorpe and S. Thrun, "Online Simultaneous Localization And Mapping with Detection And Tracking of Moving Objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2002, pp. 2918–2924.
- [41] W. Ming and S. Ji-Ying, "Simultaneous localization, mapping and detection of moving objects with mobile robot in dynamic environments," in *Proc. Int. Conf. Comp. Eng. Tech.*, April 2010, pp. 696–701.
- [42] Z. Huijing, M. Chiba, R. Shibasaki, S. Xiaowei, C. Jinshi and Z. Hongbin, "SLAM in a dynamic large outdoor environment using a laser scanner," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 1455–1462.
- [43] D. Holz, C. Lorken and H. Surmann, "Continuous 3D sensing for navigation and SLAM in cluttered and dynamic environments," in *Proc. Int. Conf. Info. Fusion.*, July 2008.
- [44] R.P.S. Mahler, "Random-set approach to data fusion," in *Proc. Automatic Object Recognition IV*. SPIE, April 1994, p. .
- [45] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA: Artech House, 2007.
- [46] B.N. Vo and S. Singh and A. Doucet, "Sequential Monte Carlo Methods for Multi-Target Filtering with Random Finite Sets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224–1245, October 2005.
- [47] B.T. Vo and B.N. Vo, "A Random Finite Set Conjugate Prior and Application to Multi-Target Tracking," in *Intel. Sensors, Sensor Networks and Info. Proc.*, December 2011, pp. 431–436.

- [48] R. Mahler, "Multitarget Bayes Filtering via First-Order Multitarget Moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, January 2004.
- [49] H. Sidenbladh, "Multi-target particle filtering for the probability hypothesis density," in *Proc. Information Fusion*, 2003, pp. 800–806.
- [50] L. Lin, Y. Bar-Shalom and T. Kirubarajan, "Track labeling and PHD filter for multitarget tracking," *IEEE Trans. Aero. Elec. Sys.*, vol. 42, no. 3, pp. 778–795, 2006.
- [51] B.N. Vo and W.K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, November 2006.
- [52] D.E. Clark and J. Bell, "Convergence results for the particle PHD filter," *IEEE Trans. Sig. Proc.*, vol. 54, no. 7, pp. 2652–2661, 2006.
- [53] D. Clark and B.N. Vo, "Convergence analysis of the gaussian mixture PHD filter," *IEEE Trans. Sig. Proc.*, vol. 55, no. 4, pp. 1204 – 1212 , 2007.
- [54] R. Mahler, "A survey of PHD filter and CPHD filter implementations," in *Proc. Sig. Process., Sensor Fusion., and Target Recog. XVI.* SPIE, April 2007, p. .
- [55] ———, "PHD Filters of Higher Order in Target Number," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1523–1543, October 2007.
- [56] B.T. Vo, B.N. Vo and A. Canton, "Analytic Implementations of the Cardinalized Probability Hypothesis Density Filter," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3553–3567, July 2007.
- [57] H. SIDENBLADH and S.L. WIRKANDER, "Tracking random sets of vehicles in terrain," in *Proc. IEEE Workshop on Multi-Object Tracking*, June 2003.
- [58] D.E. Clark, J. Bell, Y. De Saint-Pern and Y. Petillot, "PHD filter multi-target tracking in 3D sonar," in *Proc. IEEE OCEANS*, 2005.
- [59] D.E. Clark and J. Bell, "Bayesian multiple target tracking in forward scan sonar images using the PHD filter," in *IEE Proceedings Radar, Sonar and Navigation*, October 2005, pp. 327–334.
- [60] D. Clark, I.T. Ruiz, Y. Petillot and J. Bell, "Particle PHD filter multiple target tracking in sonar image," *IEEE Trans. Aeros. Elec. Sys.*, vol. 43, no. 1, pp. 409–416, January 2007.
- [61] B. Kalyan, B.A.A.P. Balasuriya and W.S. Wijesoma, "Multiple target tracking in underwater sonar images using Particle-PHD filter," in *Proc. IEEE OCEANS*, 2006.

- [62] B. Kalyan, K.W. Lee, W.S. Wijesoma, D. Moratuwage and N.M. Patrikalakis, "A random finite set based detection and tracking using 3D LIDAR in dynamic environments," in *Proc. IEEE Sys. Man. Cyber.*, October 2010.
- [63] K.W. Lee, B. Kalyan, S. Wijesoma, M. Adams, F.S. Hover and N.M. Patrikalakis, "Tracking random finite objects using 3D LIDAR in marine environments," in *Proc. ACM Symposium on Applied Computing*, 2010.
- [64] T.M. Wood, C.A. Yates, D.A. Wilkinson and G. Rosser, "Simplified Multitarget Tracking Using the PHD Filter for Microscopic Video Data," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 5, pp. 702–713, May 2012.
- [65] A. Masnadi-Shirazi and B.D. Rao, "An ICA-SCT-PHD Filter Approach for Tracking and Separation of Unknown Time-Varying Number of Sources," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 828–841, April 2013.
- [66] M. Tobias and A.D. Lanterman, "Probability hypothesis density-based multitarget tracker using multiple bistatic range and velocity measurements," in *Proc. Southeast Symp. on System Theory*, March 2004, pp. 205–209.
- [67] ———, "Probability hypothesis density-based multitarget tracking with bistatic range and Doppler observations," in *IEE Proc. Radar, Sonar, and Navig.*, vol. 152, no. 3, 2005.
- [68] ———, "Multitarget tracking using multiple bistatic range measurements with probability hypothesis densities," in *Signal Processing, Sensor Fusion, and Target Recognition XIII. Edited by Kadar, Ivan. Proceedings of the SPIE*, vol. 5429, no. , 2004, pp. 296–305.
- [69] Y.D. Wang, J.K. Wu, A.A. Kasim and W.M. Huang, "Tracking a variable number of human groups in video using probability hypothesis density," in *Proc. Int. Conf. on Pattern Recognition*, 2006, p. .
- [70] N. Ikoma, T. Uchino and H. Maeda, "Tracking of feature points in image sequence by SMC implementation of PHD filter," in *Proc. Soc. of Instrument & Contr. Engineers (SICE) Annual Conf.*, 2004, pp. 1696–1701.
- [71] J. Mullane, B.N. Vo, M.D. Adams and W.S. Wijesoma, "A Random Set Formulation for Bayesian SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, September 2008, pp. 1043–1049.
- [72] J. Mullane, B.N. Vo and M.D. Adams, "Rao-Blackwellised PHD SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 5410–5416.
- [73] J. Mullane, B.N. Vo, M.D. Adams and B.T. Vo, "A Random-Finite-Set Approach to Bayesian SLAM," *IEEE Trans. Robot. Autom.*, vol. 27, no. 2, pp. 268–282, February 2011.

- [74] ———, *Random Finite Sets for Robot Mapping & SLAM: New Concepts in Autonomous Robotic Map Representations*. Berlin: Springer, 2011.
- [75] C.S. Lee, D.E. Clark and J. Salvi, “SLAM with Single Cluster PHD Filters,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012.
- [76] M.S. Arulampalam, N. Gordon, S. Maskell and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Trans. Sig. Proc.*, vol. 50, no. 2, pp. 174–188, February 2002.
- [77] D. Schuhmacher, Ba-Tuong Vo and Ba-Ngu Vo, “A Consistent Metric for Performance Evaluation of Multi-Object Filters,” *IEEE Trans. Signal. Process.*, vol. 56, no. 8, pp. 3447–3457, August 2008.
- [78] T. Kamada and K. Olkawa, “AMADEUS: a mobile, autonomous decentralized utility system for indoor transportation,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1998.
- [79] J.W. Fenwick, P.M. Newman and J.J. Leonard, “Cooperative Concurrent Mapping and Localization,” in *Proc. IEEE Int. Conf. Robot. Autom.*, August 2002, pp. 1810–1817.
- [80] S.B. Williams, G. Dissanayake and H. Durrant-Whyte, “Towards multi-vehicle simultaneous localization and mapping,” in *Proc. IEEE Int. Conf. Robot. Autom.*, August 2002, pp. 2743–2748.
- [81] X. Zhou and S. Roumeliotis, “Multi-robot SLAM with unknown initial correspondence: the robot rendezvous case,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Sys.*, Oct 2006, pp. 1785–1792.
- [82] A. Howard, “Multi-robot simultaneous localization and mapping using Particle Filters,” *The Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1243–1256, December 2006.
- [83] L. Carlone, M.K. Ng, J. Du, B. Bona and M. Indri, “Rao-Blackwellized Particle Filters Multi Robot SLAM with Unknown Initial Correspondences and Limited Communication,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 243–249.
- [84] L.D.L. Perera and E. Nettleton, “Nonlinear Observability of the Centralized Multi-vehicle SLAM Problem,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010.
- [85] S. Thrun and Y. Liu, “Multi-robot SLAM with sparse extended information filters,” in *Proc. Int. Symp. Robot. Res.*, Sienna, Italy, October 2003, pp. 254–266.

- [86] A. Howard, G.S. Sukhatme and M.J. Mataric, "Multirobot Simultaneous Localization and Mapping Using Manifold Representations," *Proc. IEEE*, vol. 94, no. 7, pp. 1360–1369, July 2006.
- [87] A. Howard, "Multi-robot mapping using manifold representations," in *Proc. IEEE Int. Conf. Robot. Autom.*, July 2004, pp. 4198–4203.
- [88] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz and B. Stewart, "Distributed Multirobot Exploration and Mapping," *Proc. IEEE*, vol. 94, no. 7, pp. 1325–1339, July 2006.
- [89] H.J. Chang, C.S.G. Lee, Y.C. Hu and Y. Lu, "Multi-Robot SLAM with Topological/Metric Maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Nov 2007, pp. 1467–1472.
- [90] L.A.A. Andersson and J. Nygard, "C-SAM: Multi-Robot SLAM using Square Root Information Smoothing," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008.
- [91] F. Dellaert and M. Kaess, "Square root SAM: simultaneous localization and mapping via square root information smoothing," *The Int. J. Robot. Res.*, vol. 25, no. , p. 11811203, 2006.
- [92] A. Cunningham, M. Paluri and F. Dellaert, "DDF-SAM: Fully Distributed SLAM using Constrained Factor Graphs," in *Proc. IEEE Int. Conf. Intel. Robot. Sys.*, October 2010.
- [93] R. Mahler, "The multisensor PHD filter: I. General solution via multitarget calculus," in *Proc. SPIE Sig. Process., Sensor Fusion., and Target Recog. XVIII*. SPIE, April 2009, pp. 1043–1049.
- [94] M.E. Liggins, C.Y. Chong, I. Kadar, M.G. Alford, V. Vannicola and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," *Proc. of the IEEE*, vol. 85, no. 1, pp. 95–107, Jan 1997.
- [95] D. Stoyan, W.S. Kendall, and J. Mecke, *Stochastic Geometry and its Applications*. : John Wiley and Sons, Inc., 1995.
- [96] S. Nagappa and D.E. Clark, "On the ordering of the sensors in the iterated-corrector probability hypothesis density (PHD) filter," in *Signal Processing, Sensor Fusion, and Target Recognition XX*, SPIE, May 2011.
- [97] G. Kitagawa, "Monte-Carlo lter and smoother for nonGaussian nonlinear state space models," *J. of Computa. and Graphi. Stat.*, vol. 5, no. 1, pp. 1–25.
- [98] J.S. Liu, R. Chen and T. Logvinenko., "A theoretical framework for sequential importance sampling with resampling. In A. Doucet, N. de Freitas, and N. Gordon, editors. Sequential Monte Carlo Methods in Practice." *Springer-Verlog*, pp. 225–246, 2001.

- [99] A. Ess, K. Schindler, B. Leibe and L.V. Gool, "Object Detection and Tracking for Autonomous Navigation in Dynamic Environments," *The Int. J. Robot. Res.*, vol. 29, no. 14, pp. 1707–1725, December 2010.
- [100] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten and S. Teller, "An Atlas Framework for Scalable Mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2003.
- [101] M. Bosse, P. Newman, J. Leonard, and S. Teller, "Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework," *The Int. J. Robot. Res.*, vol. 23, no. 12, pp. 1113–1139, December 2004.
- [102] M. B. J.C. Leedekerken, J.J. Leonard and A. Balasuriya, "Real-time obstacle avoidance and mapping for auvs operating in complex environments," in *Proc. of the Seventh International Symposium on Technology and the Mine Problem*, May 2006.
- [103] S.A. PASHA, B.N. Vo, H.D. Tuan and W.K. Ma, "A Gaussian Mixture PHD filter for Jump Markov System Models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 3, pp. 919–936, July 2009.
- [104] C. Estrada, J. Neira and J.D. Tardos, "Finding good cycle constraints for large scale multi-robot SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009.
- [105] R. Mahler, B.T. Vo and B.N. Vo, "CPHD filtering with unknown clutter rate and detection profile," *IEEE Trans. Sig. Proc.*, vol. 59, no. 8, pp. 3497–3513, 2011.