
THE UNIVERSITY of ADELAIDE

# Methods for Online UAV Path Planning for Tracking Multiple Objects

by

**Hoa Van Nguyen**

B. Eng. (Electrical Engineering),
Portland State University, Oregon, U.S.A, 2012

Dissertation submitted for the degree of

**Doctor of Philosophy**

in

School of Computer Science
Faculty of Engineering, Computer & Mathematical Sciences
The University of Adelaide

June 2020

**Supervisors:**
Associate Professor Damith Chinthana Ranasinghe ,
School of Computer Science,
The University of Adelaide

Doctor Seyed Hamid Rezatofighi,
School of Computer Science,
The University of Adelaide

# Contents

# Abstract

Unmanned aerial vehicles (UAVs) or drones have rapidly evolved to enable carrying various sensors such as thermal sensors for vision or antennas for radio waves. Therefore, drones can be transformative for applications such as surveillance and monitoring because they have the capability to greatly reduce the time and cost associated with traditional tasking methods. Realising this potential necessitates equipping UAVs with the ability to perform missions autonomously. This dissertation considers the problems of online path planning for UAVs for the fundamental task of surveillance comprising of tracking and discovering multiple mobile objects in a scene.

Tracking and discovering an unknown and time-varying number of objects is a challenging problem in itself. Objects such as people or wildlife tend to switch between various modes of movements. Measurements received by the UAV's on-board sensors are often very noisy. In practice, the on-board sensors have a limited field of view (FoV), hence, the UAV needs to move within range of the mobile objects that are scattered throughout a scene. This is extremely challenging because neither the exact number nor locations of the objects of interest are available to the UAV.

Planning the path for UAVs to effectively detect and track multi-objects in such environments poses additional challenges. Path planning techniques for tracking a single object are not applicable. Since there are multiple moving objects appearing and disappearing in the region, following only certain objects to localise them accurately implies that a UAV is likely to miss many other objects. Furthermore, online path planning for multi-UAVs remains challenging due to the exponential complexity of multi-agent coordination problems.

In this dissertation, we consider the problem of online path planning for UAV-based localisation and tracking of multi-objects. First, we realised a low cost on-board radio receiver system on a UAV and demonstrated the capability of the drone-based platform for autonomously tracking and locating multiple mobile radio-tagged objects in field trials. Second, we devised a track-before-detect filter coupled with an online path planning algorithm for joint detection and tracking of radio-tagged objects to achieve better performance in noisy environments. Third, we developed a multi-objective planning algorithm for multi-agents to track and search multi-objects under the

practical constraint of detection range limited on-board sensors (or FoV limited sensors). Our formulation leads to a multi-objective value function that is a monotone submodular set function. Consequently, it allows us to employ a greedy algorithm for effectively controlling multi-agents with a performance guarantee for tracking discovered objects while searching for undiscovered mobile objects under practical constraints of limited FoV sensors. Fourth, we devised a fast distributed tracking algorithm that can effectively track multi-objects for a network of stationary agents with different FoVs. This is the first such solution to this problem. The proposed method can significantly improve capabilities of a network of agents to track a large number of objects moving in and out of the limited FoV of the agents' sensors compared to existing methods that do not consider the problem of unknown and limited FoV of sensors.

# Statement of Originality

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint award of this degree.

The author acknowledges that copyright of published works contained within this dissertation resides with the copyright holder(s) of those works.

I give permission for the digital version of my dissertation to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

| | |
|---|---|
| | 11 / 06 / 2020 |
| Signed | Date |

# Acknowledgements

I would like to thank my supervisors, Associate Professor Damith Chinthana Ranasinghe and Doctor Seyed Hamid Rezatofighi for their exceptional support, guidance and inspiration. It is my absolute pleasure to work with them over the last three and a half years. This dissertation would not be possible without them. The knowledge that I got from my supervisors is tremendous, and probably the most treasured asset in my life. In particular, I am sincerely grateful to be a student of A/Prof. Ranasinghe, who encouraged and transformed me from a process engineer to become an independent researcher.

Further, I would like to express my gratitude to Professor Ba-Ngu Vo for giving me the opportunity to visit him in Perth, as well as providing me guidance and invaluable feedback over the last two years. It was my great honour to collaborate with him, and I have learnt a lot from his endless knowledge.

My gratitude is extended to my lab mates, Mr. Fei Chen and Mr. Yang Su, for their tireless support in conducting field trials with a countless number of field trips, even under tough and unexpected weather conditions in Australia. I am also in debt to the Unmanned Research Aircraft Facility (URAF), the University of Adelaide for providing me tremendous support with their expertise in flying drones and drones-related compliance.

My appreciation also goes to the University of Adelaide for awarding me with the Adelaide International Scholarship (AIS), and giving me a chance to study in Adelaide—a beautiful city with friendly people. In particular, I would like to thank all of my Vietnamese friends in Adelaide, especially our traditional Friday lunch meetings which really enriched my research life and helped me through the difficulties of academic life.

My special thanks go my dear parents, Cuong and Xuan, my wife, Dung, and two beloved daughters, My and An, for always loving and encouraging me to be a better person. Their infinite support helped me to complete this research.

# Dissertation Conventions

The following conventions have been adopted in this dissertation:

## Typesetting

This document was compiled using LaTeX2e. Texstudio 2.12.22 was used as a text editor interfaced to LaTeX2e. Inkscape 0.92.3 was used to produce schematic diagrams and other drawings.

## Spelling

Australian English spelling conventions have been used, as defined in the Macquarie English Dictionary—A. Delbridge (Ed.), Macquarie Library, North Ryde, NSW, Australia, 2001.

## Referencing

The Harvard reference style is used for referencing and citation in this dissertation.

## System of Units

The units comply with the international system of units recommended in an Australian Standard: AS ISO 1000-1998 (Standards Australia Committee ME/71, Quantities, Units and Conversions 1998).

# Acronyms

| | |
|---|---|
| **CPHD** | Cardinalised Probability Hypothesis Density |
| **DTT** | Detection-Then-Track |
| **FISST** | FInite Set StatisTic |
| **FoV** | Field of View |
| **GCI** | Generalised Covariance Intersection |
| **GCM** | Global Criterion Method |
| **GCS** | Ground Control System |
| **GLMB** | Generalised Labelled Multi-Bernoulli |
| **JMS** | Jump Markov System |
| **JPDA** | Joint Probabilistic Data Association |
| **LMB** | Labelled Multi-Bernoulli |
| **MC** | Monte Carlo |
| **MHT** | Multiple Hypotheses Tracking |
| **MOP** | Multi-objective Optimisation Problem |
| **MOT** | Multi-Object Tracking |
| **MS-BF** | Multi-Sensor Bernoulli Filter |
| **PHD** | Probability Hypothesis Density |
| **POMDP** | Partially Observable Markov Decision Process |
| **OSPA** | Optimal Sub-Pattern Assignment |
| **RFS** | Random Finite Set |
| **SNR** | Signal-to-Noise Ratio |
| **RSSI** | Received Signal Strength Indicator |
| **STFT** | Short-Time Fourier Transform |
| **SDR** | Software-Defined Radio |
| **TBD** | Track-Before-Detect |
| **UAV** | Unmanned Aerial Vehicle |
| **VHF** | Very High Frequency |

# Publications

## Journal Articles

**[1] Nguyen, H.V.**, Rezatofighi, H., Vo, B.N. Ranasinghe, D.C. 2019b. Online UAV path planning for joint detection and tracking of multiple radio-tagged objects. *IEEE Transactions on Signal Processing* 67(20), pp. 5365–5379 (CORE Rank: A*, impact factor: 5.23).

**[2] Nguyen, H.V.**, Chesser, M., Koh, L.P., Rezatofighi, H. Ranasinghe, D.C. 2019a. TrackerBots: autonomous unmanned aerial vehicle for real-time localization and tracking of multiple radio-tagged animals. *Journal of Field Robotics* 36(3), pp. 617–635 (CORE Rank: A, impact factor: 4.345).

## Conference Articles

**[1] Nguyen, H.V.**, Rezatofighi, H., Vo, B.N. Ranasinghe, D.C. 2020b. Multi-objective multi-agent planning for jointly discovering and tracking mobile objects. *Proceedings of the 34th AAAI Conference on Artificial Intelligence* (CORE Rank: A*, acceptance rate: 20.6%, 1,591 accepted out of 7,737 submissions).

**[2] Nguyen, H.V.**, Chesser, M., Chen, F., Rezatofighi, H. Ranasinghe, D.C. 2018a. Demo abstract: autonomous UAV sensor system for searching and locating VHF radio-tagged wildlife. *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. ACM, pp. 333–334 (CORE Rank: A* — a demonstration paper).

**[3] Nguyen, H.V.**, Rezatofighi, H., Taggart, D., Ostendorf, B. Ranasinghe, D.C. 2018b. TrackerBots: software in the loop study of quad-copter robots for locating radio-tags in a 3D space. *Proceedings of the Australasian Conference on Robotics and Automation*. pp. 304–313.

## Under Preparation for Submissions

**[1] Nguyen, H.V.**, Chen, F., Chesser, J., Rezatofighi, H. Ranasinghe, D.C. 2020c. LAVAPilot: lightweight UAV trajectory planner with situational awareness for

embedded autonomy to track and locate radio-tags. *Submitted to the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

**[2] Nguyen, H.V.**, Rezatofighi, H., Vo, B.N., Ranasinghe, D.C. 2020a. Distributed multi-sensor multi-object tracking under limited field of view sensors. *Prepared to submit to IEEE Transactions on Signal Processing*.

# List of Figures

# List of Tables

# Chapter 1

# Introduction

This chapter introduces the motivation for the problems considered in this dissertation and discusses the challenging nature of the problems. This chapter provides a summary of the contributions made in the following chapters and provides an overview of the structural organisation of the dissertation to help readers.

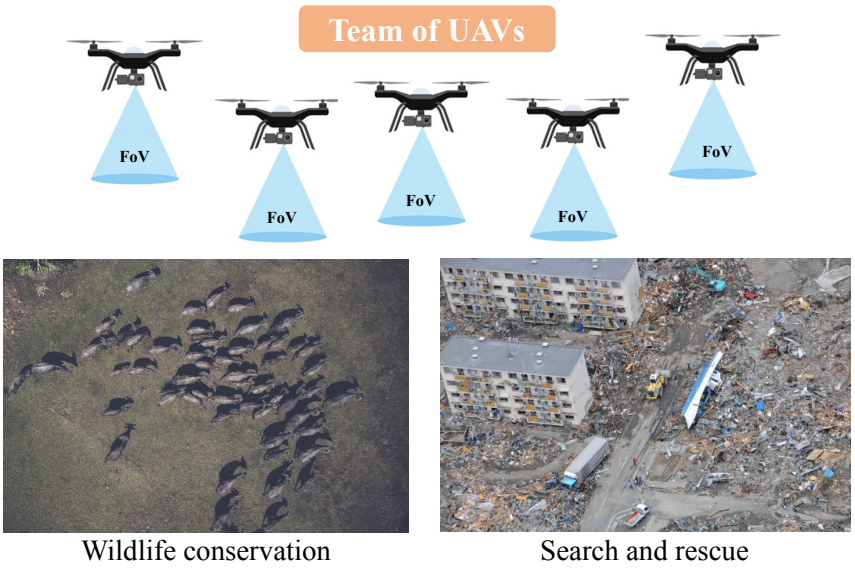


**Figure 1.1. Examples of using a team of UAVs.** UAVs can be deployed to monitor wildlife activities (left) or to search for victims during disasters (right). Each UAV comprises of a field of view (FoV) limited sensor to survey the environment and gather surveillance data.

## 1.1 Introduction

Arguably, one of the emerging disruptive technologies of the 21st century is what the Harvard Business Review (Anderson, 2017) has recently coined the "Internet of Flying Things", referring to the latest generation of consumer-grade drones or UAVs, capable of carrying imaging, thermal or even chemical/radiation/biological sensors. UAVs are touted to be transformational for tasks from wildlife monitoring, agricultural inspection, building inspection, to threat detection, as they have the potential to dramatically reduce both the time and cost associated with a traditional manual tasking based on human operators (Chung et al., 2018). Furthermore, falling unit cost prices have significantly reduced the entry barriers to employing commercial drones for solving real-world applications. However, achieving the full potential of UAVs demands us to consider autonomous UAVs for tasks.

This dissertation concentrates on a broad class of problems related to achieving autonomous UAVs (or so-called agents) with limited field of view (FoV) sensors for the task of tracking multiple mobile objects of interest in a scene. Such problems are ubiquitous in real-world applications such as wildlife tracking (Kays et al., 2011; Thomas, Holland and Minot, 2012; Cliff et al., 2015), and search and rescue missions (Gerasenko et al., 2001; Murphy et al., 2008). While a single UAV has limited competences such as FoV and endurance, employing a team of UAVs can significantly

improve the field of view, task success, and reliability. As illustrated in Figure 1.1, a team of UAVs can be deployed to monitor activities of endangered radio-tagged wildlife in a survey scene, or to search for victims in a disaster response (Beck et al., 2018). However, to realise autonomous UAVs for tracking dynamic objects in a scene requires solving two challenging problems *jointly*:

- Multi-object tracking (MOT) with limited FoV sensors,
- Online path planning for controlling UAVs to track mobile objects.

The objective of MOT is to estimate an unknown and time-varying number of object trajectories from noisy sensor measurements. MOT is a general problem for various applications such as radar (Blackman and Popoli, 1999), robotics (Mullane et al., 2011), computer vision (Cox and Hingorani, 1996), traffic monitoring (Munz, Mählisch and Dietmayer, 2010; Reuter et al., 2017), cell biology (Hoseinnezhad et al., 2012), and sensor networks (Battistelli et al., 2013). Despite development of various tracking algorithms, MOT is still a challenging problem because sensor measurements are often noisy and non-linear wherein the measurement functions of object states are usually modelled by non-linear equations with noise terms. Additionally, the noisy measurements lead to false-alarms, misdetections and data association uncertainty[1]. Further, the number of objects can be time-varying because objects can appear (birth) or disappear (death) in the surveillance area at any time (Mahler, 2007b).

For multi-agent MOT, the problems can be investigated under centralised or distributed approaches. Centralised multi-agent MOT is challenging wherein the complexities of exact solutions are combinatorial in the number of agents because of unknown data associations (*i.e.*, object to measurement assignments) that lead to combinatorial complexity when updating objects' belief densities with measurements (Mahler, 2014). Moreover, the centralised approach is prone to a single point of failure. Recently, distributed multi-agent MOT approaches have gained more interest because of its reliability, scalability and flexibility (Luo et al., 2006). However, fusing distributed information among agents is a challenging problem where optimal distributed fusion is only possible if the correlations between agents (*e.g.*, double-counting of information) are completely known (Mahler, 2000). Therefore, optimal distributed fusion can only be employed in a few restricted scenarios (*e.g.*,

[1]The data association uncertainty is the uncertainty in determining the relationship between measurements and objects, or so-called the objects-measurements assignment uncertainty.

synchronous and tree-connected networks where an oracle is assumed to constantly keep track of the full network's state) (Üney, Clark and Julier, 2013). Consequently, robust (likely sub-optimal) distributed fusion algorithms have been proposed in various literature under the assumption of sensors *without* FoV limitations (Mahler, 2000; Battistelli et al., 2013; Wang et al., 2016; Fantacci et al., 2018). The problem is increasingly more complex in case of limited FoV sensors, wherein resolving the object identities (labels) from multiple limited FoV observations of the same object leads to label or identity inconsistencies between agents, the so-called label inconsistency problems (Li et al., 2018).

Online path planning to compute optimal control actions for a team of UAVs is essential in MOT since the measurement's quality depends considerably on the travelling paths taken by the UAVs. However, computing such optimal control actions in an online manner in the context of MOT where the state (such as location) of objects cannot be directly observed presents additional challenges. In practice, the on-board sensors need to be lightweight to reduce the impact of flight times on a UAV while sensors themselves are constrained in the surveillance region by limited FoVs. Hence, it is critical to not only track the movements of discovered objects of interest but also search for undiscovered objects. Consequently, the overall team's objectives in path planning arise as a natural multi-objective optimisation problem, where several competing objectives (*i.e.*, tracking and discovering) need to be simultaneously achieved. However, multi-agent planning to achieve *multiple competing objectives* remains a challenging problem because of the complex interactions between agents leading to combinatorial optimisation problems (Wai et al., 2018). Most critically, the computation of optimal control actions must be online for MOT applications.

The aim of this dissertation is to consider the aforementioned challenging problems of UAV path planning for multi-object tracking (MOT) to realise real-world autonomous aerial robotic systems to track multiple mobile objects with noisy and limited FoV sensor measurements where an object's state cannot be directly observed by the agents.

## 1.2 Summary of Original Contributions

This dissertation comprises of several original contributions to the fields of multi-object tracking, especially with multiple-agents, and online path planning for controlling UAVs to track and locate multiple mobile objects. These are described as follows:

1. A new autonomous aerial vehicle system for simultaneously tracking and localising multiple mobile radio-tagged animals is proposed. This is based on exploiting the simplicity of *range-only* measurements. Our realisation of a lightweight UAV platform— under 2 kg—makes the technology more accessible in jurisdictions, such as Australia, where systems under 2 kg can be flown without a pilot licence. We formulate a *joint* tracking and path planning problem to realise a real-time and online autonomous system. Further, our formulation considers the trade-offs between location accuracy and resource constraints of the UAV, its manoeuvrability, and power constraints to develop a practical solution. We validate our method through extensive simulations and field experiments with mobile, very high frequency (VHF) radio-tags commonly used by conservation biologists for radio-tagging animals. To the best of our knowledge, ours is the first demonstration of an autonomous online aerial robot system for tracking and locating multiple mobile VHF radio-tags in real-time. This work has been published in the *Journal of Field Robotics* under the title of "TrackerBots: autonomous unmanned aerial vehicle for real-time localization and tracking of multiple radio-tagged animals" (Nguyen et al., 2019a).

2. The problem of online path planning for joint detection and tracking of multiple unknown radio-tagged objects is considered. We propose an online path planning algorithm with joint detection and tracking because signal measurements from these objects are inherently noisy. We derive a partially observable Markov decision process (POMDP) with a random finite set (RFS) track-before-detect (TBD) multi-object filter, which also maintains a safe distance between the UAV and the objects of interest using a void probability constraint. We show that, in practice, the multi-object likelihood function of raw signals received by the UAV in the time-frequency domain is separable and results in a numerically efficient multi-object TBD filter. We derive a TBD filter with a jump Markov system (JMS) to accommodate manoeuvring objects capable of switching between different dynamic modes. Our evaluations demonstrate the capability of the proposed approach to handle multiple radio-tagged objects subject to birth, death, and motion modes. Moreover, this online planning method with the TBD-based filter outperforms its detection-based counterparts in detection and tracking, especially in low signal-to-noise ratio (SNR) environments. This work has been published in the *IEEE Transactions on Signal Processing* under the

title "Online UAV path planning for joint detection and tracking of multiple radio-tagged objects" (Nguyen et al., 2019b).

3. A framework for multiple agents to jointly plan, search and track a time-varying number of objects using a novel multi-objective information-based value function formulation is proposed. Our multi-objective value function captures the competing objectives of planning for *tracking* and *discovery*. We adopt the RFS model for the collection of objects of interest to account for the random appearance and disappearance of objects and their dynamics. Our proposed multi-objective value function maximises information gain over a look-ahead horizon for both discovered and undiscovered objects. Most importantly, our multi-objective value function is proven to be a monotone submodular set function; thus, we can cope with the intractability of the multi-objective optimisation problem (MOP) by employing a greedy algorithm. Our ability to use a greedy algorithm facilitates the computation of approximately optimal control actions with linear complexity in the number of agents for realising an online planning method. This work has been accepted by *The 34th AAAI Conference on Artificial Intelligence* under the title "Multi-objective multi-agent planning for jointly discovering and tracking mobile objects" (Nguyen et al., 2020b).

4. A fast and efficient fusion algorithm in distributed multi-sensor networks with limited computational resources for tracking multiple objects under limited FoV sensors is developed. In particular, a novel fusion strategy is conducted on local multi-object trajectory estimates instead of local multi-object densities. As a result, the proposed fusing algorithm is significantly faster than traditional densities-based fusion algorithms. Moreover, the tracking accuracy is improved by directly examining the multi-object tracking errors using the optimal sub-pattern assignment (OSPA) metric. Experimental results demonstrate that our proposed method can perform in real-time and substantially outperforms the state-of-the-art fusion rules such as generalised covariance intersection (GCI) in terms of speed and accuracy in challenging scenarios with limited FoV sensors. This work is prepared to submit to the *IEEE Transactions on Signal Processing* under the title "Distributed multi-sensor multi-object tracking under limited field of view sensors" (Nguyen et al., 2020a).

| | | |
|---|---|---|
| **Introduction** | *Chapter 1* | • Introduction.<br>• Motivations and our contributions. |
| | *Chapter 2* | • Background on tracking and planning. |
| **Single-UAV** | *Chapter 3* | • RSSI-based aerial platform to track radio-tagged wildlife.<br>• A light weight, low cost payload system.<br>• Autonomous planning algorithm for a UAV to improve tracking accuracy and conserve on-board power. |
| | *Chapter 4* | • Joint detection and tracking of unknown and time-varying number of radio-tagged objects.<br>• Derive an efficient multi-object TBD filter to track objects under noisy environments.<br>• Situational awareness with an online planning method under a TBD-based filter. |
| **Multi-UAV** | *Chapter 5* | • A team of agents to search and track a time-varying number of mobile objects in centralised manner.<br>• A multi-objective value function for online planning implemented using a greedy algorithm with performance guarantee. |
| | *Chapter 6* | • A network of distributed stationary agents or so-called nodes to track a time-varying and unknown number of mobile objects.<br>• Derive a lightweight and efficient fusion algorithm for agents with different field of views. |
| **Conclusion** | *Chapter 7* | • Conclusion.<br>• Research outlook – future work. |

**Figure 1.2. Outline of the dissertation.**

## 1.3 Dissertation Structure

The dissertation structure is outlined in Figure 1.2, and is described as follows.

1. Chapter 1 and Chapter 2 provides a brief introduction and background in tracking and planning tasks. We also discuss the challenges and opportunities of using autonomous UAVs to track multiple objects under limited FoV sensors.

2. Chapter 3 focuses on developing a received signal strength indicator (RSSI)-based aerial platform to track radio-tagged wildlife. A low cost and lightweight payload system that can be mounted on a commercial UAV is developed. Additionally, an autonomous and online planning algorithm for a UAV under a POMDP framework to track multiple radio-tagged objects is implemented and verified through extensive field experiments.

3. Chapter 4 considers a challenging problem of online UAV planning for tracking unknown and time-varying number of radio-tagged objects under noisy environments. An efficient TBD filter for on-off-keying signals used in radio-tagged transmitters is derived. Additionally, a JMS is implemented with the TBD filter to accommodate manoeuvring objects capable of switching between different dynamic modes. Furthermore, an online planning method for the UAV using a void probability functional for situational awareness to maintain a safe distance to objects of interest is investigated.

4. Chapter 5 investigates a problem of searching and tracking multiple objects using multiple UAVs under range-limited on-board sensors. A multi-objective value function considers both tracking and searching objectives simultaneously is proposed and proved to be a monotone set function. Hence, a greedy algorithm to control multiple UAVs in a centralised manner can be implemented with a performance bound.

5. Chapter 6 examines a problem of tracking a time-varying number of mobile objects using a network of distributed stationary agents or so-called nodes. This work derives an efficient and fast fusion algorithm for distributed nodes under practical constraints of unknown data association measurements and limited FoV sensors.

6. Chapter 7 gives a conclusive summary for all of the completed work in this dissertation, and discuss the potential work in the future.

Chapter 2

# Fundamentals of Tracking and Planning

Our applications are based on the problems of online path planning for UAVs to track an unknown and time-varying number of objects. In this chapter, we provide a general concept of single-object and multi-object tracking under the Bayesian paradigm. Further, we provide a brief background on online path planning for UAVs under the partially observable Markov decision process (POMDP) framework. We also provide the definitions for the metrics used to quantify our results.

## 2.1 Notations

For notational consistency, we use lowercase letters ( *e.g.*, $x$) for single-object states; capital letters (*e.g.*, $X$) represent the multi-object states; bold letters (*e.g.*, $\boldsymbol{x}, \boldsymbol{X}$) are used for labelled states; blackboard letters (*e.g.*, $\mathbb{X}$) denote state spaces. Let $1_A(\cdot)$ denote the inclusion function of a given set $A$, and $\mathcal{F}(A)$ denote the class of finite subset of $A$. If $X = \{x\}$, for convenience, write $1_A(x)$ instead of $1_A(\{x\})$. For simplicity, albeit with a slight abuse of notation, we use the symbol $f(\cdot|\cdot)$ to denote the single-object and multi-object transition kernels, and the symbol $g(\cdot|\cdot)$ to denote the single-object and multi-object measurement likelihood functions.

## 2.2 Bayesian Filtering

The Bayesian filtering approach is an online estimation method dealing with the problem of inferring knowledge about the unobserved state of a dynamic system, which changes over time, from a sequence of noisy observations. In a standard Bayesian filter when the number of objects is fixed and known, the object states and observations are commonly modelled as random vectors with fixed dimensions. Formally, suppose $x_k$ is the state of an object at time $k$, which generates an observation $z_k$ based on the observation model:

$$z_k = h_k(x_k, w_k), \tag{2.1}$$

where $w_k$ denotes the observation noise. In general, the observation can be characterised by a likelihood function $g_k(z_k|x_k)$, which is the probability of observing the measurement $z_k$ given the state $x_k$. Further, the object state $x_k$ evolves over time based on the transition model:

$$x_k = f_{k-1}(x_{k-1}, v_{k-1}), \tag{2.2}$$

where $v_{k-1}$ denotes the process noise. Generally, the object state can also be characterised by the transition kernel $f_{k|k-1}(x_k|x_{k-1})$, which is the probability of transitioning to the object state $x_k$, given its previous state $x_{k-1}$.

The objective of the filtering problem is to estimate the belief density $\pi_k(x_k|z_{1:k})$ based on the history of observation data $z_{1:k}$ from time 1 to time $k$. Using the Bayes recursion roles, from the initial density $\pi_0$, the belief density can be calculated sequentially using

the prediction and update steps as follows:

$$\pi_{k|k-1}(x_k|z_{1:k-1}) = \int f_{k|k-1}(x_k|x)\pi_{k-1}(x|z_{1:k-1})dx, \tag{2.3}$$

$$\pi_k(x_k|z_{1:k}) = \frac{g_k(z_k|x_k)\pi_{k|k-1}(x_k|z_{1:k-1})}{\int g_k(z_k|x)\pi_{k|k-1}(x|z_{1:k-1})dx}. \tag{2.4}$$

The prediction step (2.3) is also called a Chapman-Kolmogorov equation, while the update step (2.4) is a well-known Bayes rule.

### 2.2.1 Kalman filters and beyond

Kalman filter (KF), firstly introduced in 1960 by Rudolph E. Kalman in (Kalman, 1960), is the simplest form of the Bayesian filters, when states and measurement variables are linear and have Gaussian distributions, given by:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + q_{k-1}, \\ z_k &= H_k x_k + r_k. \end{aligned} \tag{2.5}$$

Here, $q_{k-1} \sim \mathcal{N}(0, Q_{k-1})$ is the process noise and $r_k \sim \mathcal{N}(0, R_k)$ is the measurement noise, $F_{k-1}$ is the state transition matrix and $H_k$ is the observation matrix with the initial distribution is Gaussian $x_0 \sim \mathcal{N}(m_0, P_0)$, such that:

$$\begin{aligned} f_{k|k-1}(x_k|x_{k-1}) &= \mathcal{N}(x_k; F_{k-1}x_{k-1}, Q_{k-1}), \\ g_k(z_k|x_k) &= \mathcal{N}(z_k; H_k x_k, R_k). \end{aligned} \tag{2.6}$$

Assume at $k-1$ time step, the state belief density is:

$$\pi_{k-1}(x_{k-1}|z_{1:k-1}) = \mathcal{N}(x_{k-1}; m_{k-1}, P_{k-1}). \tag{2.7}$$

Substituting (2.7) and (2.6) into (2.3) results in the predicted belief density at time $k$,

$$\pi_{k|k-1}(x_k|z_{1:k-1}) = \mathcal{N}(x_k; m_{k|k-1}, P_{k|k-1}), \tag{2.8}$$

where $m_{k|k-1} = F_{k-1}m_{k-1}, P_{k|k-1} = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$.

Substitute (2.8) and (2.6) into (2.4) results in the filtering belief density at time $k$:

$$\begin{aligned} \pi_k(x_k|z_{1:k}) &= \frac{g_k(z_k|x_k)\pi_{k|k-1}(x_k|z_{1:k-1})}{\int g_k(z_k|x)\pi_{k|k-1}(x|z_{1:k-1})dx}, \\ &\propto \mathcal{N}(z_k|H_k x_k, R_k)\mathcal{N}(x_k|m_{k|k-1}, P_{k|k-1}), \\ &\propto \mathcal{N}(x_k|m_k, P_k), \end{aligned} \tag{2.9}$$

where

$$\begin{aligned} m_k &= m_{k|k-1} + P_{k|k-1}H_k^T(H_k P_{k|k-1} H_k^T + R_k)^{-1}(z_k - H_k m_{k|k-1}), \\ &= m_{k|k-1} + K_k(z_k - H_k m_{k|k-1}), \\ P_k &= P_{k|k-1} - P_{k|k-1}H_k^T(H_k P_{k|k-1} H_k^T + R_k)^{-1} H_k P_{k|k-1}, \\ &= (I - K_k H_k) P_{k|k-1}, \\ S_k &= R_k + H_k P_{k|k-1} H_k^T, \\ K_k &= P_{k|k-1} H_k^T S_k^{-1}. \end{aligned} \tag{2.10}$$

Here, $K_k$ is called a *Kalman gain*, $(z_k - H_k m_{k|k-1})$ is called an *innovation*, and $S_k$ is called an *innovation covariance*.

**Beyond Kalman filters:** Although Kalman filter is the optimal solution for linear problems with Gaussian distributions, it cannot be implemented in several real world scenarios because most practical filtering problems are non-linear, non-Gaussian problems. Consequently, a more general scenario of non-linear systems is often considered, *i.e.*,

$$\begin{aligned} x_k &= f_{k-1}(x_{k-1}) + q_{k-1}, \\ z_k &= h_k(x_k) + r_k. \end{aligned} \tag{2.11}$$

When dynamic and measurement models are somewhat non-linear, two popular approximation solutions are extended Kalman filter (EKF) (Bar-Shalom, 1987; Jazwinski, 2007) and unscented Kalman filter (UKF) (Julier, Uhlmann and Durrant-Whyte, 2000; Julier and Uhlmann, 2004).

*Extended Kalman Filter:* The EKF filter applies Taylor series to approximate the non-linear function $f_{k-1}(\cdot)$ and $h_k(\cdot)$ in (2.11) where the Taylor series is usually pruned at the first linear term, so-called first-order EKF filter. By linearisation, the EKF filter has the same form as KF filter, wherein the transition function $f_{k-1}(\cdot)$ and measurement function $h_k(\cdot)$ are replaced by its corresponding Jacobian matrix $\hat{F}_{k-1}$ and $\hat{H}_k$, given by

$$\hat{F}_{k-1} = \frac{\partial f_{k-1}}{\partial x}\Big|_{x=m_{k-1}}, \qquad \hat{H}_k = \frac{\partial h_k}{\partial x}\Big|_{x=m_{k|k-1}}. \tag{2.12}$$

*Unscented Kalman Filter:* Unlike the EKF filter which approximates the $f_{k-1}(\cdot)$ and $h_k(\cdot)$ functions through Taylor series expansion, the UKF filter instead captures the mean and covariance of the filtering belief distribution $\pi_k(x_k|z_{1:k})$ through the *deterministically* chosen sample points (or so-called sigma points) (Julier, Uhlmann and Durrant-Whyte, 2000). For an $n$-dimensional state variable with assumed Gaussian

distribution: $x_{k-1} \sim \mathcal{N}(m_{k-1}, P_{k-1})$, form a set of $2n+1$ sigma points $\mathcal{X}^{(i)}$ with corresponding weight $W^{(i)}$ for $i^{\text{th}}$ point as follow:

$$\begin{aligned}
\mathcal{X}_{k-1}^{(0)} &= m_{k-1},\ W^{(0)} = \kappa/(n+\kappa), \\
\mathcal{X}_{k-1}^{(i)} &= m_{k-1} + \left(\sqrt{(n+\kappa)P_{k-1}}\right)_i,\ W^{(i)} = 1/2(n+\kappa), \\
\mathcal{X}_{k-1}^{i+n} &= m_{k-1} - \left(\sqrt{(n+\kappa)P_{k-1}}\right)_i,\ W^{i+n} = 1/2(n+\kappa),
\end{aligned} \tag{2.13}$$

where $\kappa \in \mathbb{R}$, $\left(\sqrt{(n+\kappa)P_{k-1}}\right)_i$ is the $i$th row of the matrix square root $L$ of $(n+\kappa)P_{k-1}$ such that $(n+\kappa)P_{k-1} = L^T L$.

Using the sampled set in (2.13), the UKF filter procedure follows:

- Propagate predicted sigma points: $\mathcal{X}_{k|k-1}^{(i)} = f(\mathcal{X}_{k-1}^{(i)}), i = 0, 1, 2, ..., 2n$.
- Compute predicted mean and covariance:

$$m_{k|k-1} = \textstyle\sum_{i=0}^{2n} W^{(i)} \mathcal{X}_{k|k-1}^{(i)}, \quad P_{k|k-1} = \sum_{i=0}^{2n} W^{(i)} (\mathcal{X}_{k|k-1}^{(i)} - m_{k|k-1})(\mathcal{X}_{k|k-1}^{(i)} - m_{k|k-1})^T + Q_{k-1}.$$

- Compute predicted measurement: $\hat{z}_{k|k-1} = \sum_{i=0}^{2n} W^{(i)} h_k(\mathcal{X}_{k|k-1}^{(i)})$.
- Compute updated mean and covariance using actual measurements:

$$m_k = m_{k|k-1} + K_k(z_k - \hat{z}_{k|k-1}), \quad P_k = P_{k|k-1} - P_{xz} K_k^T,$$

where

$$K_k = P_{xz} S_k^{-1}, \quad S_k = R_k + P_{zz}, P_{xz} = \sum_{0}^{2n} W^{(i)} (\mathcal{X}_{k|k-1}^{(i)} - m_{k|k-1})(h_k(\mathcal{X}_{k|k-1}^{(i)}) - \hat{z}_{k|k-1})^T,$$

$$P_{zz} = \sum_{0}^{2n} W^{(i)} (h_k(\mathcal{X}_{k|k-1}^{(i)}) - \hat{z}_{k|k-1})(h_k(\mathcal{X}_{k|k-1}^{(i)}) - \hat{z}_{k|k-1})^T.$$

### 2.2.2 Particle filters

The particle filters (PF) (Gordon, Salmond and Smith, 1993; Gordon, 1997; Doucet, De Freitas and Gordon, 2001; Ristic, Arulampalam and Gordon, 2004) belong to a class of approximation methods to non-linear systems in the Bayesian filter family. The basic method of the particle filters is to use a random sampling process (Monte Carlo) to approximate the probability distributions of interest (Vo et al., 2015). The particle filters implement the random sampling process called Monte Carlo (MC) method to

approximate the belief density by a weighted set of independently and identically distributed (i.i.d) particles.

Suppose at time $k-1$, the belief density is approximated by a set of particles, $\{(w_{k-1}^{(i)}, \tilde{x}_{k-1}^{(i)})\}_{i=1}^{N}$, *i.e.*, :

$$\pi_{k-1}(x_{k-1}|z_{1:k}) \approx \sum_{i=1}^{N} w_{k-1}^{(i)} \delta(x_{k-1} - \tilde{x}_{k-1}^{(i)}), \tag{2.14}$$

where $\delta(\cdot)$ denotes the Kronecker delta, and $\sum_{i=1}^{N} w_{k-1}^{(i)} = 1$.

Each particle is predicted to time $k$ using the dynamic transition model, given by:

$$\tilde{x}_{k|k-1}^{(i)} = f_{k-1}(\tilde{x}_{k-1}^{(i)}) + q_{k-1}, \tag{2.15}$$

while the weight $w_{k|k-1}^{(i)} = w_{k-1}^{(i)}$ is maintained the same during the prediction step.

In contrast, during the update step, the particle state is maintained, *i.e.*, $\tilde{x}_{k}^{(i)} = \tilde{x}_{k|k-1}^{(i)}$, while its corresponding weight is updated, given by

$$w_{k}^{(i)} = \frac{g_k(z_k|\tilde{x}_{k}^{(i)}) w_{k|k-1}^{(i)}}{\sum_{i=1}^{N} g_k(z_k|\tilde{x}_{k}^{(i)}) w_{k|k-1}^{(i)}}. \tag{2.16}$$

A typical problem of particles filter is particles depletion, *i.e.*, weights are concentrated on a few particles, while the rest of the particles have weights closed to zero. The reason is that, after update procedures, the variance of weights increases and never decreases because the measurement likelihood function is often less scattered than the dynamic transition kernel. A well-known method to prevent particle depletion is called resampling, which can be done in various ways, such as multimodal resampling, systematic resampling, soft systematic resampling, bootstrap filter resampling, regularised particle filter and auxiliary particle filter (Doucet, De Freitas and Gordon, 2001; Ristic, Arulampalam and Gordon, 2004; Douc and Cappé, 2005; Hol, Schon and Gustafsson, 2006). The main idea is that the particles with very small weights are pruned, while the very high weight particles are copied.

## 2.3 Random Finite Sets

The previous subsection provides most well-known filters for tracking single-object, wherein only one object and one measurement exist. For multi-object tracking,

several heuristic approaches have been proposed using an array of single-object trackers such as global nearest neighbour (GNN) (Blackman and Popoli, 1999), joint probabilistic data association (JPDA) (Bar-Shalom, 1987), multi-hypotheses tracking (MHT) (Reid, 1979). However, these algorithms involve an explicit object to measurement assignment (data association), which becomes less reliable during high false-alarms or objects in closed space. Further, these trackers require a known and fixed number of objects since their underlying states are random variables, which have fixed and known dimensions. Note that even for cases of wildlife tracking, when the data association problem is solved via unique radio-tagged identity resulting an array of single-object trackers can be used for MOT; however, the existence of objects is difficult to confirm, especially under low SNR conditions. As a result, there are remaining critical applications in which the number of objects is unknown and time-varying. A more suitable model is random finite set (RFS) framework, an emerging paradigm that generalises the classical dynamical systems to set-valued dynamical systems, in which MOT is a multi-object state estimation problem (Mahler, 2007b).

### 2.3.1 Definition

An RFS $X$ on $\mathbb{X}$ is a random variable taking values in the finite subsets of $\mathbb{X}$. Using Mahler's finite set statistic (FISST), an RFS is fully described by its FISST density. The FISST density is not a probability density (Mahler, 2007b), but it is equivalent to a probability density as shown in (Vo, Singh and Doucet, 2005). We introduce three common RFSs, Bernoulli RFS, multi-Bernoulli RFS and labelled multi-Bernoulli RFS used in our work.

### 2.3.2 Bernoulli RFS

A Bernoulli RFS $X$ on $\mathbb{X}$ has at most one element with probability $r$ for being a singleton distributed over the state space $\mathbb{X}$ according to PDF $p(\cdot)$, and probability $1-r$ for being empty. Its FISST density is defined as follows (Mahler, 2007b, pp. 351):

$$\pi(X) = \begin{cases} 1-r & X = \emptyset, \\ r \cdot p(x) & X = \{x\}, \end{cases} \tag{2.17}$$

while its cardinality distribution $\rho(\cdot)$ is a Bernoulli distribution parameterised by $r$.

### 2.3.3 Multi-Bernoulli RFS

A Multi-Bernoulli RFS is a union of a fixed number (say $N$) of independent Bernoulli RFSs: $X = \bigcup_{i=1}^{N} X^{(i)}$, where $X^{(i)}$ is a Bernoulli RFS on $\mathcal{F}(\mathbb{X})$ characterised by the existence probability $r^{(i)}$ and probability density $p^{(i)}$ defined on $\mathbb{X}$. Its FISST density is given by (Mahler, 2007b, pp. 368):

$$\pi(\{x^{(1)},\ldots,x^{(n)}\}) = \pi(\emptyset) \sum_{1\leq i_1\neq\cdots\neq i_n\leq N} \prod_{j=1}^{n} \frac{r^{(i_j)}\cdot p^{(i_j)}(x^{(j)})}{1-r^{(i_j)}}, \tag{2.18}$$

where $\pi(\emptyset) = \prod_{i=1}^{N}(1-r^{(i)})$, and its cardinality distribution is also a multi-Bernoulli distribution (Mahler, 2007b, pp. 369):

$$\rho(n) = \pi(\emptyset) \sum_{1\leq i_1<\cdots<i_n\leq N} \prod_{j=1}^{n} \frac{r^{(i_j)}}{1-r^{(i_j)}}. \tag{2.19}$$

### 2.3.4 Labelled multi-Bernoulli RFS

A labelled RFS with state space $\mathbb{X}$ and label space $\mathbb{L}$ is an RFS on $\mathbb{X}\times\mathbb{L}$ where all realisations of labels are distinct. Similar to the multi-Bernoulli RFS, a labelled multi-Bernoulli (LMB) RFS is completely defined by a parameter set $\{(r^{(\lambda)}, p^{(\lambda)}) : \lambda\in\Psi\}$ with index set $\Psi$. Its FISST density is given by (Reuter et al., 2014):

$$\boldsymbol{\pi}(\boldsymbol{X}) = \delta_{|\boldsymbol{X}|}(|\mathcal{L}(\boldsymbol{X})|)w(\mathcal{L}(\boldsymbol{X}))p^{\boldsymbol{X}}, \tag{2.20}$$

where $\delta$ is the Kronecker delta, $\mathcal{L}(\boldsymbol{X})$ denotes the set of labels extracted from $\boldsymbol{X}\in\mathcal{F}(\mathbb{X}\times\mathbb{L})$, $p(\boldsymbol{x}) = p(x,\lambda) = p^{(\lambda)}(x)$, $p^{\boldsymbol{X}} = \prod_{(x,\lambda)\in\boldsymbol{X}} p^{(\lambda)}(x)$, $w(L) \triangleq \prod_{i\in\mathbb{L}}(1-r^{(i)})\prod_{\lambda\in L}\frac{1_{\mathbb{L}}(\lambda)r^{(\lambda)}}{(1-r^{(\lambda)})}$.

### 2.3.5 Generalised labelled multi-Bernoulli RFS

A generalised labelled multi-Bernoulli (GLMB) RFS with state space $\mathbb{X}$ and label space $\mathbb{L}$ is an RFS on $\mathbb{X}\times\mathbb{L}$ such that its FISST density follows (Vo and Vo, 2013):

$$\boldsymbol{\pi}(\boldsymbol{X}) = \triangle(\boldsymbol{X}) \sum_{\xi\in\Xi} w^{(\xi)}[p^{(\xi)}]^{\boldsymbol{X}}, \tag{2.21}$$

where $\triangle(\boldsymbol{X}) \triangleq \delta_{|\boldsymbol{X}|}(|\mathcal{L}(\boldsymbol{X})|)$ is the distinct label indicator, $\Xi$ is a discrete space, $p^{(\xi)}(\cdot,l)$ is a probability density on $\mathbb{X}$, and $w^{(\xi)}(I)$ is a non-negative hypothesis weight with

$\sum_{I\subseteq\mathbb{L}}\sum_{\xi\in\Xi} w^{(\xi)}(I) = 1$. An alternative form of the GLMB is known as $\delta$-GLMB, which is used to facilitate the numerical implementation. A $\delta$-GLMB RFS represents the statistical dependencies among objects by considering multiple hypotheses comprising of a set of track labels $I \in \mathbb{L}$ and a corresponding association history $\xi \in \Xi$. Its FISST density is given by (Vo, Vo and Phung, 2014):

$$\boldsymbol{\pi}(\boldsymbol{X}) = \triangle(\boldsymbol{X}) \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)}\delta_I(\mathcal{L}(\boldsymbol{X}))[p^{(\xi)}]^{\boldsymbol{X}}. \tag{2.22}$$

Note that the $\delta$-GLMB is realised from the GLMB using the relationship $w^{\xi}(J) = \sum_{I\in\mathcal{F}(\mathbb{L})} w^{(\xi)}(I)\delta_I(J)$.

## 2.4 Multi-object Filtering using RFS Theory

In the FISST approach, the multi-object state at time $k$ is modelled as a (labelled) RFS $\boldsymbol{X}_k$. The representation of a multi-object state by a finite set provides consistency with the notion of estimation error distance (Vo et al., 2010). Let $z_{1:k}$ denote the history of measurement data from time 1 to $k$. Then using the FISST concept of density and integration, the filtering densities can be propagated using the prediction and update steps of the Bayes multi-object filter (Mahler, 2007b):

$$\boldsymbol{\pi}_{k|k-1}(\boldsymbol{X}_k|z_{1:k-1}) = \int \boldsymbol{f}_{k|k-1}(\boldsymbol{X}_k|\boldsymbol{X}_{k-1})\boldsymbol{\pi}_{k-1}(\boldsymbol{X}_{k-1}|z_{1:k-1})\delta\boldsymbol{X}_{k-1}, \tag{2.23}$$

$$\boldsymbol{\pi}_k(\boldsymbol{X}_k|z_{1:k}) = \frac{g(z_k|\boldsymbol{X}_k)\boldsymbol{\pi}_{k|k-1}(\boldsymbol{X}_k|z_{1:k-1})}{\int g(z_k|\boldsymbol{X})\boldsymbol{\pi}_{k|k-1}(\boldsymbol{X}|z_{1:k-1})\delta\boldsymbol{X}}, \tag{2.24}$$

where $\boldsymbol{\pi}_{k|k-1}(\cdot|z_{1:k-1})$ denotes a multi-object predicted density; $\boldsymbol{\pi}_k(\cdot|z_{1:k})$ denotes a multi-object filtering density; $\boldsymbol{f}_{k|k-1}(\cdot|\cdot)$ denotes a transition kernel from time $k-1$ to $k$; $g(z_k|\cdot)$ denotes a measurement likelihood function at time $k$. Note that the multi-object transition kernel $\boldsymbol{f}_{k|k-1}(\cdot|\cdot)$ incorporates all dynamic aspects of objects including death, birth and transition to new states. The integral is a *set integral* defined for any function $\boldsymbol{p} : \mathcal{F}(\mathbb{X}\times\mathbb{L}) \to \mathbb{R}$, given by:

$$\int \boldsymbol{p}(\boldsymbol{X})\delta\boldsymbol{X} = \sum_{n=0}^{\infty}\frac{1}{n!}\sum_{(l_1,\ldots,l_n)\in\mathbb{L}^n}\int_{\mathbb{X}^n} p(\{(x^{(1)},l^{(1)}),\ldots,(x^{(n)},l^{(n)})\})d(x^{(1)},\ldots,x^{(n)}). \tag{2.25}$$

Generally, the FISST Bayes multi-object recursion is intractable. However, considerable interest in the field has lead to a number of filtering solutions such as the probability

hypothesis density (PHD) filter (Mahler, 2003), the cardinalised PHD (CPHD) filter (Mahler, 2007a), the multi-object multi-Bernoulli (MB) filter (Mahler, 2007b; Vo, Vo and Cantoni, 2009), the generalised labelled multi-Bernoulli (GLMB) filter (Vo and Vo, 2013; Vo, Vo and Phung, 2014), and the labelled multi-Bernoulli (LMB) filter (Reuter et al., 2014).

## 2.5 Path Planning for UAVs

Given this dissertation considers the problem of path planning for multiple UAVs, we provide a brief background in path planning theories. The prominent planning algorithms such as genetic algorithm (GA) and particle swarm optimisation (PSO) (Roberge, Tarbouchi and Labonte, 2013) are not applicable for online applications because it is computationally expensive due to bio-inspired approaches. As shown in (Kaelbling, Littman and Cassandra, 1998), the online planning problem is similar to the problem of an agent computing optimal actions under a partially observable Markov decision process (POMDP) to maximise its reward, which has gained significant interest recently (Baek et al., 2013; Ragi and Chong, 2013). Additionally, at the theoretical level, the POMDP framework facilitates direct generalisation to MOT using RFS (Mahler, 2007b). This can be called as RFS-POMDP — a POMDP with the information state being the filtering density of the RFS of objects.

Multi-agent path planning in partially observable environments is further challenging due to the complex coordination among agents. Although the cooperation problem can be formulated as a decentralised POMDP (Dec-POMDP), its exact solutions are NEXP-hard (Bernstein et al., 2002). This is especially problematic for multi-agent POMDPs since the action and observation space grows exponentially with the number of agents (Amato and Oliehoek, 2015). One approach is distributed POMDPS (*e.g.*, networked distributed POMDP (Nair et al., 2005)) by exploiting interactions among neighbouring agents using distributed constraint optimisation. However, realising a global goal for multi-agents in a distributed manner is an NEXP-problem in worst case scenarios (Rizk, Awad and Tunstel, 2018). Hence, to cope with this intractability, the MPOMDP centralised approach (Messias, Spaan and Lima, 2011) can be adopted for controlling multiple agents (Dames and Kumar, 2015; Dames, Tokekar and Kumar, 2017; Wang et al., 2018).

Therefore, this section provides a brief background on POMDP for planning a single agent (*e.g.*, a single UAV) and MPOMDP for planning multiple agents (*e.g.*, multiple UAVs).

### 2.5.1 Partially observable Markov decision process (POMDP)

POMDP (partially observable Markov decision process) is a theoretical framework for stochastic control problems and is described by the 6-tuple $\left[\mathcal{F}(\mathbb{T}) \times \mathbb{U}, \mathbb{A}, \mathcal{T}, \mathcal{R}, \mathcal{F}(\mathbb{Z}), g(\cdot|\cdot)\right]$ where (Monahan, 1982; Lovejoy, William S, 1991; Bertsekas et al., 1996; Hsu, Lee and Rong, 2008):

- $\mathbb{T} = \mathbb{X} \times \mathbb{L}$ is the labelled state space;
- $\mathcal{F}(\mathbb{T}) \times \mathbb{U}$ is the space where each of its elements is an ordered pair $(\boldsymbol{X}, u)$, with $\boldsymbol{X}$ being an object state (possibly a multi-object state) and $u$ an observer state;
- $\mathbb{A}$: a set of control actions;
- $\mathcal{T}$: a state-transition function on $\left[\mathcal{F}(\mathbb{T}) \times \mathbb{U}\right] \times \mathbb{A} \times \left[\mathcal{F}(\mathbb{T}) \times \mathbb{U}\right]$ where $\mathcal{T}((\boldsymbol{X}, u), a, (\boldsymbol{X}', u'))$ is the probability density of next state $(\boldsymbol{X}', u')$ given current state $(\boldsymbol{X}, u)$ and action $a$ taken by the observer;
- $\mathcal{R}$: a real-valued reward function defined on $\mathbb{A}$;
- $\mathcal{F}(\mathbb{Z})$: a set of observations;
- $g(\cdot|\cdot)$: an observation likelihood function on $\mathcal{F}(\mathbb{Z}) \times \left[\mathcal{F}(\mathbb{T}) \times \mathbb{U}\right] \times \mathbb{A}$ where $g(z|(\boldsymbol{X}, u), a)$ is the likelihood of an observation $z$ given the state $(\boldsymbol{X}, u)$, after the observer takes the action $a$.

The main goal in a POMDP is to find an optimal action $a_k^*$ that generates an optimal trajectory (a sequence of observer's positions) by maximising the total expected reward over $H$ look-ahead steps. Specifically, the total expected reward is $\mathbb{E}[\sum_{j=1}^{H} \gamma^{j-1} \mathcal{R}_{k+j}(a_k)]$ with $\mathbb{E}[\cdot]$ denoting the expectation operator, and discount factor $\gamma \in (0, 1]$ to moderate the effects of future rewards on current actions.

In this dissertation, we propose using an information-based reward function. For the purpose of joint detection and tracking, where reducing overall uncertainty[2]

[2]The uncertainty associated with the estimated state of all objects.

is the main objective, such a reward function is more appropriate because more information implies less uncertainty (Beard et al., 2017). There are other reward functions, such as cardinality variances (Gostar, Hoseinnezhad and Bab-Hadiashar, 2013), which are good at estimating the number of objects or the OSPA-based method in (Gostar et al., 2017) that depends on user-defined threshold values. In contrast, the information-based methods capture overall cardinality and position information, and can be efficiently computed in a closed-form. A detailed comparison between task-based and information-based reward functions can be found in (Kreucher, Hero and Kastella, 2005).

Suppose $\boldsymbol{\pi}_{k+H|k}(\cdot|z_{1:k})$ is the predicted density to time $k+H$ given measurements up to time $k$, which can be calculated recursively by only using the Bayes prediction step in (2.23) from time $k$ to $k+H$. Now, suppose $a_k$ is the control action applied to the UAV at time $k$; then, the UAV follows a trajectory consisting of a sequence of discrete positions $u_{k+1:k+H}(a_k) = [u_{k+1}(a_k), \ldots, u_{k+H}(a_k)]^T$ with corresponding hypothesised measurements $z_{k+1:k+H}(a_k) = [z_{k+1}(a_k), \ldots, z_{k+H}(a_k)]^T$. Then the filtering density $\boldsymbol{\pi}_{k+H}(\cdot|z_{1:k}, z_{k+1:k+H}(a_k))$ can be computed recursively using the Bayes filter in (2.23) and (2.24) from time $k$ to $k+H$. The reward function can be specified in terms of information divergence between the filtering density and the predicted density. The rationale is that a more informative filtering density yields better estimation results. Thus, it is appropriate to choose an optimal policy that generates a more informative filtering density. Since the filtering density is equally or more informative than the predicted density, maximising the information divergence between the filtering density and the predicted density often results in a more informative filtering density, and consequently, a better tracking performance. In particular, the information-based reward function is given by (Beard et al., 2015):

$$\mathcal{R}_{k+H}(a_k) = D(\boldsymbol{\pi}_{k+H}(\cdot|z_{1:k}, z_{k+1:k+H}(a_k)), \boldsymbol{\pi}_{k+H|k}(\cdot|z_{1:k})), \tag{2.26}$$

where $D(\boldsymbol{\pi}_2, \boldsymbol{\pi}_1)$ is the information divergence between two FISST densities, $\boldsymbol{\pi}_2$ and $\boldsymbol{\pi}_1$. Some information divergence candidates are Rényi divergence (including Kullback-Leibler divergence) and Cauchy-Schwarz divergence, described below:

**Rényi divergence** between any two FISST densities, $\boldsymbol{\pi}_2$ and $\boldsymbol{\pi}_1$, is defined as (Ristic and Vo, 2010):

$$D_{\text{Rényi}}(\boldsymbol{\pi}_2, \boldsymbol{\pi}_1) = \frac{1}{\alpha - 1} \log \int \boldsymbol{\pi}_2^{\alpha}(\boldsymbol{X}) \boldsymbol{\pi}_1^{1-\alpha}(\boldsymbol{X}) \delta\boldsymbol{X}, \tag{2.27}$$

where $\alpha \geq 0$ is a parameter which determines the emphasis of the tails of two distributions in the metric. When $\alpha \to 1$, we obtain the well-known Kullback-Leibler (KL) divergence.

**Cauchy-Schwarz divergence** between any two FISST densities, $\pi_2$ and $\pi_1$, is defined as (Hoang et al., 2015):

$$D_{CS}(\pi_2, \pi_1) = -\log\left(\frac{\int K^{|X|}\pi_2(\boldsymbol{X})\pi_1(\boldsymbol{X})\delta\boldsymbol{X}}{\sqrt{\int K^{|X|}\pi_2^2(\boldsymbol{X})\delta\boldsymbol{X}\int K^{|X|}\pi_1^2(\boldsymbol{X})\delta\boldsymbol{X}}}\right), \tag{2.28}$$

where $K$ denotes the unit of hyper-volume on $\mathbb{T}$.

### 2.5.2 Multi-agent POMDP (MPOMDP)

Multi-agent POMDP is a centralised control framework for multiple agents wherein each agent shares its observations via communications to a centralised controller. Let $\mathcal{F}(A)$ denote the class of finite subsets of $A$. An MPOMDP is described by a tuple $\left[S, H, \mathcal{F}(\mathbb{X}) \times \mathbb{U}^S, \mathbb{A}^S, \mathcal{F}(\mathbb{Z})^S, \mathcal{T}, \mathcal{R}, \mathcal{O}\right]$ where

- $S$ is the number of agents;
- $H$ is the look-ahead horizon;
- $\mathcal{F}(\mathbb{X}) \times \mathbb{U}^S$ is the space, wherein each element is an ordered pair $(X, U)$, with $X$ is the object state and $U = [u^1, \ldots, u^S]^T \in \mathbb{U}^S$ is states of $S$ agents;
- $\mathbb{A}^S = \mathbb{A} \times \cdots \times \mathbb{A}$ is the control action space for $S$ agents resulting in the joint action $A = [a_1, \ldots, a^S]^T \in \mathbb{A}^S$;
- $\mathcal{F}(\mathbb{Z})^S = \mathcal{F}(\mathbb{Z}) \times \cdots \times \mathcal{F}(\mathbb{Z})$ is the space of joint observations resulting in the joint observation $Z = [Z^1, \ldots, Z^S]^T \in \mathcal{F}(\mathbb{Z})^S$;
- $\mathcal{T} : \left[\mathcal{F}(\mathbb{X}) \times \mathbb{U}^S\right] \times \left[\mathcal{F}(\mathbb{X}) \times \mathbb{U}^S\right] \times \mathbb{A}^S \to [0, 1]$ defines the transition probabilities $\Pr\big((X', U')|(X, U), A\big)$;
- $\mathcal{R} : \left[\mathcal{F}(\mathbb{X}) \times \mathbb{U}^S\right] \times \mathbb{A}^S \to \mathbb{R}$ defines the immediate reward of performing action $A$ in state $\left[\mathcal{F}(\mathbb{X}) \times \mathbb{U}^S\right]$;
- $\mathcal{O} : \mathcal{F}(\mathbb{Z})^S \times \left[\mathcal{F}(\mathbb{X}) \times \mathbb{U}^S\right] \times \mathbb{A}^S \to [0, 1]$ defines the joint observation probabilities $\Pr\big(Z|(X', U'), A\big)$.

## 2.6 Performance Evaluation Metrics

In this section, we present the key evaluation metrics used in the multi-object tracking and employed to evaluate the performance of the algorithms developed herein. These include: i) root mean square error (RMSE), ii) optimal sub-pattern assignment (OSPA), and iii) OSPA-on-OSPA (OSPA$^{(2)}$).

### 2.6.1 Root mean square error (RMSE)

When the number of object is known and unchanged over time, RMSE is often used to measure the estimation error between the estimated value versus its ground truth.

Let $\hat{X} = [\hat{x}_1, \ldots, \hat{x}_n]^T \in \mathbb{X}$ be an estimate of its ground truth $X = [x_1, \ldots, x_n]^T \in \mathbb{X}$ where $n$ is the number of objects (or so-called cardinality). The RMSE is the square root of the mean square error, given by:

$$d_{\text{rmse}} = \sqrt{\frac{\sum_{i=1}^{n} \left[d(\hat{x}_i, x_i)\right]^2}{n}}, \tag{2.29}$$

where $d(\hat{x}_i, x_i)$ is a single object metric (*e.g.*, Euclidean distance) on $\mathbb{X}$.

### 2.6.2 Optimal sub-pattern assignment (OSPA)

When the number of objects is unknown and time-varying, RMSE is not suitable since it does not cover the miss-distance concept, *e.g.*, the number of estimated objects is not the same as the number of true objects. Therefore, OSPA (Schuhmacher, Vo and Vo, 2008) — a new mathematically and intuitively consistent metric — is introduced to address the miss-distance concept.

Let $\mathcal{F}(\mathbb{X})$ be the space of finite subsets of $\mathbb{X}$. As stated in (Schuhmacher, Vo and Vo, 2008), let $d_p^{(c)}(X, Y)$ be the OSPA distance between $X, Y \in \mathcal{F}(\mathbb{X})$ with order $p$ and cut-off $c$. Let $m$ be the cardinality of $X = \{x^{(1)}, \ldots, x^{(m)}\}$ and $n$ be the cardinality of $Y = \{y^{(1)}, \ldots, y^{(n)}\}$. If $m \leq n$, $d_p^{(c)}(X, Y)$ is defined as

$$d_p^{(c)}(X, Y) = \left( \frac{1}{n} \left( \min_{\pi \in \Pi_n} \sum_{i=1}^{m} \bar{d}^{(c)}(x^{(i)}, y^{(\pi(i))})^p + c^p(n - m) \right) \right)^{1/p}, \tag{2.30}$$

where $\Pi_n$ is a set of all permutations of $n$ ($|\Pi_n| = n!$), $\bar{d}^{(c)}(x,y) = \min(c, d(x,y))$, in which $d(\cdot,\cdot)$ is an arbitrary metric on the single object state space of $\mathbb{X}$. If $m > n$, then $d_p^{(c)}(X,Y) \triangleq d_p^{(c)}(Y,X)$.

The OSPA distance is comprised of two components: OSPA *localisation* $\bar{d}_{p,\text{loc}}^{(c)}$ and OSPA *cardinality* $\bar{d}_{p,\text{card}}^{(c)}$ to account for localisation and cardinality errors. These components are given by with $m \leq n$:

$$\bar{d}_{p,\text{loc}}^{(c)}(X,Y) = \left( \frac{1}{n} \min_{\pi \in \Pi_n} \sum_{i=1}^{m} d^{(c)}(x^i, y^{\pi(i)})^p \right)^{1/p}, \tag{2.31}$$

$$\bar{d}_{p,\text{card}}^{(c)}(X,Y) = \left( \frac{c^p(n-m)}{n} \right)^{1/p}, \tag{2.32}$$

and if $m > n$, then $\bar{d}_{p,\text{loc}}^{(c)}(X,Y) \triangleq \bar{d}_{p,\text{loc}}^{(c)}(Y,X)$, $\bar{d}_{p,\text{card}}^{(c)}(X,Y) \triangleq \bar{d}_{p,\text{card}}^{(c)}(Y,X)$.

### 2.6.3 OSPA$^{(2)}$ metric

In multi-object tracking, the actual interest is object trajectories instead of object states over time, which involves trajectory labels (Vo and Vo, 2013). The OSPA metric only measures the distance between estimated versus true states, thus it does not consider other tracking criteria such as track fragmentation and track switching. As a result, a new adaption of OSPA, called OSPA$^{(2)}$ is invented to measure the difference between two sets of tracks (Beard, Vo and Vo, 2017, 2018; Beard, Vo and Vo, 2020). In particular, OSPA$^{(2)}$ metric is the same as the OSPA metric, except its *base distance* is itself another OSPA-based distance.

Let $\mathbb{T} = \{1, \ldots, K\}$ be a finite space of time indices, from the start time at 1 to the end time at $K$. Let $\mathbb{U} \triangleq \{f : \mathbb{T} \to \mathbb{X}\}$ be a space of all functions from $\mathbb{T}$ to $\mathbb{X}$. An element of $\mathbb{U}$ is defined as a track.

**Base distance:** Let $f, g \in \mathbb{U}$ be two tracks of $\mathbb{U}$, let $D_f, D_g \in \mathbb{T}$ be the time support of track $f$ and track $g$, respectively. The base distance between two tracks is defined as the average OSPA distance over its time support, *i.e.*,

$$\tilde{d}^{(c)}(f,g) = \begin{cases} \displaystyle\sum_{t \in D_f \cup D_g} \frac{d^{(c)}(\{f(t)\}, \{g(t)\})}{|D_f \cup D_g|}, & \text{if } |D_f \cup D_g| \neq 0, \\ 0, & \text{if } |D_f \cup D_g| = 0. \end{cases} \tag{2.33}$$

where $\{f(t)\}$ is a singleton if $t \in D_f$, and empty set otherwise(likewise for {g(t)}), $d^{(c)}(\cdot,\cdot)$ is the OSPA distance defined in (2.30), which becomes

$$d^{(c)}(\{f(t)\},\{g(t)\}) = \begin{cases} 0, & \text{if } |(\{f(t)\}| = |\{g(t)\}| = 0, \\ c, & \text{if } |(\{f(t)\}| \neq |\{g(t)\}|, \\ \min(c, d(f(t), g(t))), & \text{if } |(\{f(t)\}| = |\{g(t)\}| = 1. \end{cases} \tag{2.34}$$

**OSPA$^{(2)}$ for Tracks:** The base distance defined in (2.34) is a metric, and confined by the cut-off value $c$. Thus, it can be used as a base distance for the original OSPA distance. Let $X = \{f^{(1)},\ldots,f^{(m)}\} \in \mathcal{F}(\mathbb{U})$ and $Y = \{g^{(1)},\ldots,g^{(n)}\} \in \mathcal{F}(\mathbb{U})$ be two sets of tracks, $\check{d}_p^{(c)}(X,Y)$ be the OSPA$^{(2)}$ distance between $X$ and $Y$. If $m \leq n$, $\check{d}_p^{(c)}(X,Y)$ is defined based on the base distance $\tilde{d}^{(c)}(\cdot,\cdot)$ as

$$\check{d}_p^{(c)}(X,Y) = \left(\frac{1}{n}\left(\min_{\pi\in\Pi_n}\sum_{i=1}^{m}\tilde{d}^{(c)}(f^{(i)},y^{(\pi(i))})^p + c^p(n-m)\right)\right)^{1/p}, \tag{2.35}$$

where c is the cut-off and p is the order parameters. If $m > n$, then $\check{d}_p^{(c)}(X,Y) \triangleq \check{d}_p^{(c)}(Y,X)$.

As shown in (Beard, Vo and Vo, 2020), OSPA$^{(2)}$ is a metric with all properties including identity, symmetry, non-negative and triangle inequality.

Chapter 3

# Autonomous UAV for Tracking Multiple Radio-Tagged Objects

WE consider the problem of building an autonomous aerial robot capable of planning its trajectory to track and locate multiple mobile objects within the field of view of its sensor. We ground our problem in the application of the system for tracking and locating radio-tagged wildlife in conservation biology. We present a novel autonomous aerial vehicle system—*TrackerBots*—to track and localise multiple radio-tagged animals. Such a robot can provide new possibilities to study the habitats and behaviours of endangered species through the efficient gathering of location information at temporal-spatial granularity not possible with traditional manual survey methods. The simplicity of measuring the RSSI values of VHF radio-collars commonly used in radio-tagging is exploited to realise a low cost and lightweight sensor platform suitable for integration with UAVs. Due to uncertainty and the non-linearity of the system based on RSSI measurements, our joint tracking and planning approaches integrate a particle filter for tracking and localising and a POMDP for dynamic path planning. This approach allows autonomous navigation of a UAV in a direction of maximum information gain to locate multiple mobile animals and reduce exploration time, and, consequently, conserve on-board battery power. We validated our real-time and online approach through both extensive simulations and field experiments with five VHF radio-tags on a grassland plain.

## 3.1 Motivation and Contribution

Understanding basic questions of ecology such as how animals use their habitat, their movements and activities are necessary for addressing numerous environmental challenges ranging from invasive species to diseases spread by animals and saving endangered species from extinction. Conservation biologists, ecologists as well as natural resource management agencies around the world rely on numerous methods to monitor animals. Traditional methods using radio-tagging species of interest (Cochran and Lord Jr, 1963; Kenward, 2000) as well as more recent vision-based sensors (Selby, Corke and Rus, 2011; Olivares-Mendez et al., 2015) or infrared (IR) based sensors (Zhou, 2013; Christiansen et al., 2014; Gonzalez et al., 2016; Ward et al., 2016) are employed for these tasks. IR-based sensors are sensitive to environmental temperature and become less reliable when they are used outdoors, especially during daytime in summer months (Zhou, 2013). In general, vision-based approaches are less effective when animals are camouflaged and are susceptible to visual occlusions, e.g. by grass, shrubs and even other animals. Most significantly, due to the difficulty of automatically recognising individual animals using vision/IR based approaches, tracking multiple animals with these sensors requires dealing with the very challenging problem of data association (Bar-Shalom, 1987; Stone et al., 2013). Often, conservation biologists need tools to track and monitor a specific set of individual animal species; for example, individuals of a reintroduction species into a natural habitat. This becomes difficult to achieve in the presence of occlusions and data associations problems of IR/vision based approaches. Thus, capturing and collaring concerned species with very high frequency (VHF) radio tags and the subsequent use of VHF telemetry or radio tracking is the most important and cost-effective tool employed to study the movement of a wide range of animal sizes (Wikelski et al., 2007) in their natural environments (Kays et al., 2011; Thomas, Holland and Minot, 2012; Tremblay et al., 2017; Webber et al., 2017).

However, the traditional method of radio tracking is not without its problems. Tracking radio-collared animals typically requires researchers to trek long distances in the field, armed with cumbersome VHF radio receivers with hand-held antennas and battery packs to manually home in on radio signals emitted from radio-tagged or collared animals. Consequently, the precious spatial data acquired through radio tracking come at a significant cost to researchers in terms of manpower, time and funding. The problem is often compounded by other challenges, such as low animal

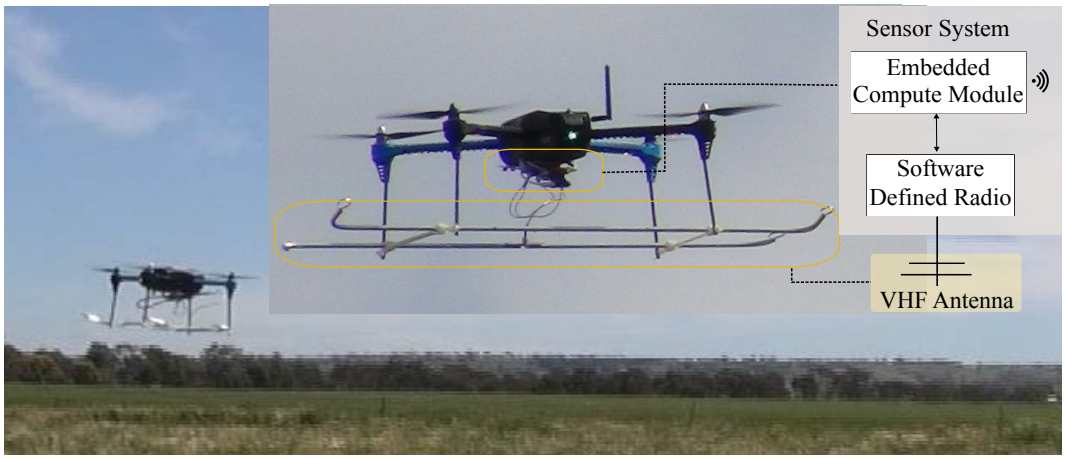


**Figure 3.1. TrackerBots.** An overview of the UAV tracking platform with its sensor system.

recapture rates, equipment failures, and the inability to track animals that move into inaccessible terrains. Furthermore, many of our most endangered species also happen to be the most difficult to track due to their small size, inconspicuousness, and location in remote habitats.

Automated tracking and location of wildlife with autonomous unmanned aerial vehicles (UAVs) can provide *new* possibilities to better understand ecology and our native wildlife to safeguard biodiversity and manage our natural resources cost-effectively. We present a low-cost approach capable of realisation in a lightweight payload for transforming existing commodity drone platforms into autonomous aerial vehicle systems as shown in Figure 3.1 to empower conservation biologists to track and localise multiple radio-tagged animals.

The main contribution of this chapter is a new autonomous aerial vehicle system for simultaneously tracking and localising multiple mobile radio-tagged animals using VHF radio-collars, commonly used in the field by conservation biologists.

## 3.2 Related Work

Our problem is embedded in the development of a UAV planning method for tracking multiple mobile radio-tagged objects using the simplicity of received signal strength measurements. Therefore: *i)* we review studies in the field of received signal strength measurement based tracking with a specific focus on methods developed for UAVs and wildlife tracking; and *ii)* we focus on related work in the field of tracking radio-collared animals using UAVs.

**Received signal strength indicator (RSSI)-based Tracking:** This method is studied in localising objects in both indoor and outdoor environments. The approach relies on using the strength of a radio signal from an emitter captured by a receiver to estimate, for example, the distance to the emitter. Related methods with possible applications to wildlife tracking can be found in the use of wireless sensor networks (WSN) for tracking a radio wave emitter. In (Caballero et al., 2008; Särkkä, Viikari and Jaakkola, 2014) a mobile beacon is localised by a fixed number of sensor nodes with known locations. The first automated VHF telemetry measurement system was reported in (Kays et al., 2011). A set of six ground-based antenna arrays deployed in a rainforest localised radio-tagged animal locations using bearing estimates obtained from signal strength measurements made by ground-based stations. These methods are advantageous for meeting long-term monitoring needs. However, the scale of the fixed and powered infrastructure required prior to a tracking task and the cost of deployment and maintenance over a large area make these approaches difficult for general use cases. In contrast, a UAV based measurement method can provide greater flexibility and a lower cost approach. Off-line estimations of a radio-tag's location obtained from signal strength data logged from a UAV was demonstrated in (Jensen, Geller and Chen, 2014). Developments in software-defined radios (SDRs) have enabled new capabilities to process multiple radio-tag signals simultaneously. Early efforts to demonstrate the possibility of incorporating SDR architectures with a UAV to detect multiple transmitted signals from radio-tags were reported in (Dos Santos et al., 2014; VonEhr et al., 2016). Notably, the studies above with UAVs were performed under the assumption of stationary radio tags. The task of autonomously tracking and locating multiple mobile radio-tagged objects from a UAV remains.

**UAV-based Autonomous Localisation and Tracking:** Since this application is related to locating VHF collared animals, we will focus on progress made towards the autonomous localisation and tracking of multiple VHF radio-tagged animals here.

Pioneering achievements in autonomous wildlife tracking have been made through simulation studies (Posch and Sukkarieh, 2009) and experimentally demonstrated systems (Körner et al., 2010; Tokekar et al., 2010; Vander Hook, Tokekar and Isler, 2014; Cliff et al., 2015) in recent years. In particular, the first demonstration of a UAV was presented in (Cliff et al., 2015).

The recent approaches (Vander Hook, Tokekar and Isler, 2014; Cliff et al., 2015) for real-time localisation of a static object (assuming stationary wildlife) used wireless

signal characteristics captured by a narrowband receiver to estimate location; in particular, the angle-of-arrival (AoA) of a radio beacon was determined using an array of antennas with the information related to a ground-based receiver for location estimations. Although the approach can conveniently manage topological variations in terrain, AoA systems require a large bulky receiver system and multiple antenna elements as well as long observation times, 45 seconds per observation as reported in (Cliff et al., 2015). Moreover, the antenna systems being mounted on top of the UAV (Cliff et al., 2015) is likely to lead to difficulty in tracking terrestrial animals although being suitable for locating avian species dwelling in trees.

**Summary:** We can see that there are few investigations that have studied the problem of locating radio-collared animals using autonomous robots. Although a system based on the angle-of-arrival was recently evaluated to be able to locate a stationary radio-collared animal, the development of a low-cost and lightweight autonomous system, capable of long-range flights and of locating multiple *mobile* radio-collared animals, still remains.

We present an alternative approach, exploiting RSSI-based measurements, because of the ability to use a simpler sensing system onboard commodity UAVs, to realise lower cost, and longer flight time, UAVs for tracking and localising multiple animals. Together with a theoretical framework for joint tracking and planning, we design, build and demonstrate a lightweight autonomous aerial robot platform. Our robot platform has the potential to provide a cost-effective method for wildlife conservation and management. To the best of our knowledge, ours is the first demonstration of an autonomous online aerial robot system for tracking and locating multiple mobile VHF radio-tags in real-time.

## 3.3 Tracking and Planning Problem Formulation

Real-time tracking requires an online estimator and a dynamic planning method. This section presents our tracking and localising formulation under the theoretical frameworks of a Bayesian filter for tracking and POMDP for planning strategy.

### 3.3.1 Tracking and localising

For tracking, we use a Bayesian filter. It is an online estimation technique which deals with the problem of inferring knowledge about the unobserved state of a dynamic system—in our problem, wildlife—which changes over time, from a sequence of noisy measurements. Suppose $x \in \mathbb{X}$ and $z \in \mathbb{Z}$ are respectively the system (kinematic) state vector in the state space $\mathbb{X}$ and the measurement (observation) vector in the observation space $\mathbb{Z}$. The problem is estimating the state $x \in \mathbb{X}$ from the measurement $z \in \mathbb{Z}$ or calculating the marginal posterior distribution $\pi_k(\cdot|z_{1:k})$ sequentially through ***prediction*** (3.1) and ***update*** (3.2) steps.

$$\pi_{k|k-1}(x_k|z_{1:k-1}) = \int f_{k|k-1}(x_k|x)\pi_{k-1}(x|z_{1:k-1})dx, \tag{3.1}$$

$$\pi_k(x_k|z_{1:k}) = \frac{g(z_k|x_k)\pi_{k|k-1}(x_k|z_{1:k-1})}{\int g(z_k|x)\pi_{k|k-1}(x|z_{1:k-1})dx}. \tag{3.2}$$

where $f_{k|k-1}(\cdot|\cdot)$ denotes a state transition kernel from time $k-1$ to time $k$, and $g(z_k|\cdot)$ denotes a measurement likelihood at time $k$.

In the case of a non-linear system or non-Gaussian noise, there is no general closed-form solution for the Bayesian recursion and $\pi_k(\cdot|z_{1:k})$ generally has a non-parametric form. Therefore, in our problem, we use a particle filter implementation as an approximate solution for the Bayesian filtering problem due to our highly non-linear measurement model.

**Particle Filter (PF):** A particle filter uses a sampling approach to represent the non-parametric form of the posterior density $\pi_k(\cdot|z_{1:k})$. The samples from the distribution are represented by a set of particles; each particle has a weight assigned to represent the probability of that particle being sampled from the probability density function. Then, these particles representing the non-parametric form of $\pi_k(\cdot|z_{1:k})$ are propagated over time. In the simplest version of the particle filter, known as the bootstrap filter first introduced by Gordon in (Gordon, Salmond and Smith, 1993), the samples are directly generated from the transitional dynamic model. Then, to reduce the particle degeneracy, resampling and injection techniques are implemented; a detailed algorithm can be found in (Ristic, Arulampalam and Gordon, 2004). See Section 2.2.2 for more details.

**Measurement model:** The update process of a PF requires the derivation of a likelihood of measurements. In our problem, based on estimating an object's—VHF radio tag's—range from the receiver, we require a realistic signal propagation model

to obtain the likelihood of receiving a given measurement. We employ two VHF signal propagation models suitable for describing RSSI measurements in non-urban outdoor environments (Jakes, 1974; Patwari et al., 2005). Denoting $h(x,u)$ as the RSSI measurement function between object $x$ and observer (UAV) state $u$, we have:

***i) Log Distance Path Loss Model (LogPath)*:** The received power is the only line of sight power component transmitted from a transmitter subjected to signal attenuation such as through absorption and propagation loss (Patwari et al., 2005):

$$h(x,u) = P_r^{d_0} - 10n\log_{10}(d(x,u_p)/d_0) + G_r(x,u), \tag{3.3}$$

where

- $x = [p_x^t, p_y^t, p_z^t]^T$ is the object's position; $u_p = [p_x^u, p_y^u, p_z^u]^T$ is the observer's (UAV) position in Cartesian coordinates; $u = [u_p; \theta^u]$ is the UAV's state which includes its heading angle $\theta^u$;
- $d(x,u_p)$ is the Euclidean distance between the object's position and UAV's position;
- $G_r(x,u)$ is the UAV receiver antenna gain which depends on its heading, its position, and object's position (details explained in Section 3.6.2);
- $P_r^{d_0}$ is received power at a reference distance $d_0$;
- $n$ is the path-loss exponent that characterises the signal losses such as absorption and propagation losses and this parameter depends on the environment with typical values ranging from 2 to 4 (Patwari et al., 2005).

***ii) Log Distance Path Loss Model with Multi-Path Fading (MultiPath)*:** The received power is composed of both line of sight power component transmitted from a transmitter and the multi-path power component reflected from the ground plane subjected to signal attenuation such as through absorption and propagation loss: (Jakes, 1974, pp. 81):

$$h(x,u) = P_r^{d_0} - 10n\log_{10}(d(x,u_p)/d_0) + G_r(x,u) + 10n\log_{10}(|1+\Gamma(\psi)e^{-j\triangle\varphi}|), \tag{3.4}$$

where, in addition to terms in (3.3),

- $\psi$ is the angle of incidence between the reflected path and the ground plane;
- $\Gamma(\psi) = [\sin(\psi) - \sqrt{\varepsilon_g - \cos^2(\psi)}]/[\sin(\psi) + \sqrt{\varepsilon_g - \cos^2(\psi)}]$ is the ground reflection coefficient with $\varepsilon_g$ is the relative permittivity of the ground;
- $\triangle\varphi = 2\pi\triangle d/\lambda$ is the phase difference between two waves where $\lambda$ is the wavelength and $\triangle d = ((p_x^t - p_x^u)^2 + (p_y^t - p_y^u)^2 + (p_z^t + p_z^u)^2)^{1/2} - d(x,u)$.

In non-urban environments, received power is usually corrupted by environmental noise, with the assumption that the noise is white, the total received power $z = P_r(x,u)$ [dBm] is:

$$z = h(x,u) + \eta_P, \tag{3.5}$$

where $\eta_P \sim \mathcal{N}(0, \sigma_P^2)$ is Gaussian white noise with covariance $\sigma_P^2$. Notably, even if RSSI noise is not completely characterised by a white noise model, we found it practical to characterise the received noise with a white Gaussian model as shown in Figure 3.7.

We use data captured in experiments using our sensor system to validate the physical sensor characteristics $G_r(x,u)$ (see Section 3.4) and $n$ defined by environmental characteristics, as well as estimate the propagation model reference power parameter $P_r^{d0}$ and noise $\sigma_P$ (see Section 3.6.2).

***Measurement likelihood*:** Based on (3.5) with Gaussian noise $\eta_P$, the likelihood of measurement $z_k$, given object and sensor position are $x_k$ and $u_k$, respectively, at time $k$ is

$$g(z_k|x_k,u_k) \sim \mathcal{N}(z_k; h(x_k,u_k), \sigma_P^2), \tag{3.6}$$

where $\mathcal{N}(\cdot;\mu,\sigma^2)$ is a normal distribution with mean $\mu$ and covariance $\sigma^2$.

### 3.3.2 Path planning

The UAV planning problem is similar to the problem of an agent computing optimal actions under a partially observable Markov decision process (POMDP) to maximise its reward. (Kaelbling, Littman and Cassandra, 1998) have shown that a POMDP framework implements an efficient and *optimal* approach based on previous actions and observations to determine the true world states. A POMDP in conjunction with a particle filter provides a principled approach for evaluating planning decisions to realise an autonomous system for tracking.

In general, a POMDP is described by the 6-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \mathcal{Z})$ where $\mathcal{S}$ is a set of both UAV and object states ($s = \{x, u\} \in S$), $\mathcal{A}$ is a set of UAV actions, $\mathcal{T}$ is a state-transition function $\mathcal{T}(s, a, s') = p(s'|s, a)$ for a given action $a$, $\mathcal{R}(a)$ is a reward function, $\mathcal{O}$ is a set of observations and $\mathbb{Z}$ is an observation likelihood $\mathbb{Z}(s, a, o) = p(o|s, a)$ with $s, s' \in \mathcal{S}$ is the current state and next state respectively, $a \in \mathcal{A}$ is the taken action and $o \in \mathcal{O}$ is the observation—i.e., measurement. The goal of a POMDP is to find an optimal policy to maximise the total expected reward $\mathbb{E}[\sum_{t=k+1}^{k+H} \gamma^{t-k} \mathcal{R}_t(a_k)]$ where $H$ is look-ahead horizon steps, $\gamma$ is the discount factor which serves as the value difference between the current reward versus the future reward, $a_k$ is action at time step $k$ and $\mathbb{E}[\cdot]$ is the expectation operator (Hsu, Lee and Rong, 2008).

The reward function can be calculated using different methods such as task-driven or information-driven strategies. When uncertainty is high, the information gain approach is preferable to reduce an object's location uncertainty (Beard et al., 2017); hence, we used this method to calculate our reward function. There are several approaches to evaluate information gain in robotic path planning such as Shannon entropy (Cliff et al., 2015), Kullback-Leibler (KL) divergence or Rényi divergence (Hero, Kreucher and Blatt, 2008). We adopted the approach in (Ristic and Vo, 2010; Ristic, 2013) to implement Rényi divergence as our reward function since it fits naturally with our Monte Carlo sampling method. Here, Rényi divergence measures the information gain between prior and posterior densities (Ristic and Vo, 2010; Beard et al., 2017):

$$\mathcal{R}_{k+H}^{(m)}(a_k) = \frac{1}{\alpha - 1} \log \int \left[\pi_{k+H|k}(x|z_{1:k})\right]^{\alpha} \left[\pi_{k+H}(x|z_{1:k}, z_{k+1:k+H}^{(m)}(a_k))\right]^{1-\alpha} dx, \quad (3.7)$$

where $\alpha \geq 0$ is a scale factor to decide the effect from the tails of two distributions. The prior density $\pi_{k+H|k}(\cdot|z_{1:k})$ is calculated by propagating current posterior particles sampled from $\pi_k(\cdot|z_{1:k})$ to time $k + H$ using the prediction step (3.1). The posterior density $\pi_{k+H}(\cdot|z_{1:k}, z_{k+1:k+H}^{(m)}(a_k))$ where $z_{k+1:k+H}^{(m)}(a_k)$ is the *future* measurement set that will be observed if action $a_k \in \mathcal{A}_k$ is taken; this is calculated by applying both prediction (3.1) and update steps (3.2) up to time $k + H$. However, using Bayes update procedure is computationally expensive and prohibitive in a real-time setting. Instead, we implement a computationally efficient approach using a black box simulator proposed in (Silver and Veness, 2010) along with the Monte Carlo sampling approach. Hence, the problem transforms to find an optimal action $a_k^* \in \mathcal{A}_k$ to maximise total

expected reward:

$$a_k^* \approx \arg\max_{a_k \in \mathcal{A}_k} \frac{1}{M_s} \sum_{t=k+1}^{k+H} \sum_{m=1}^{M_s} \gamma^{t-k} \mathcal{R}_t^{(m)}(a_k), \tag{3.8}$$

where $M_s$ is the number of future measurements.

### 3.3.3 Multi-object tracking

The particle filter proposed in Section 3.3.1 can be extended to multi-object tracking (MOT). However, MOT normally deals with the complex data association problem where it is difficult to determine which measurement belongs to which object. In contrast, for our system, each object can be estimated from the measurement based on the signal frequency and tracked independently. Thus, we do not need to solve the data association problem. Notably, not all of the objects are detected due to, for example, the UAV movements, the measurement range limits imposed by propagation losses and receiver sensitivity. Therefore, if the object is not detected, the solver does not update its estimated position.

Besides maximising the number of objects localised and tracked, we formulated a termination condition for each object to conserve UAV battery power; an object is considered localised if its location uncertainty, determined by a determinant of its particles covariance, is sufficiently small ($< N_{Th}$). Note that the tracking accuracy can be improved if the uncertainty threshold $N_{Th}$ is tightened, however, tightening $N_{Th}$ may lead to a longer flight time since the UAV needs to spend more time to localise objects. Further, those found objects are *forgotten* to aid the solver to prioritise its computing resources of the ground control system on those objects with high uncertainty. The the proposed termination condition ensure the greedy planning strategy—*trajectory planning to reduce the uncertainty of the closest target*—does not consider unnecessary UAV actions to further reduce the estimation uncertainty of already localised targets and waste UAV battery power.

## 3.4 System Implementation

We implemented an experimental aerial robot system based on our tracking and planning formulation. An overview of the complete system is described in Figure 3.2a.

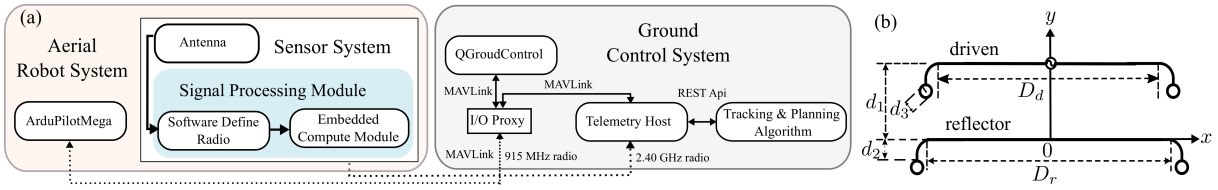


**Figure 3.2. TrackerBots' system implementation.** a) The communication channels between UAV and the ground control system (GCS) with its main software components and protocols. The solid lines represent the internal connections/communications within the sensor system and the GCS. The dotted lines are connections between wireless interfaces such as the aerial robot system and the GCS through two different radio channels: 915 MHz and 2.4 GHz. b) The folded 2-element Yagi antenna design used in our sensor system for observations.

Our experimental system used a 3DR IRIS+ UAV platform and a new sensor system built with: *i)* a compact directional VHF antenna design, and *ii)* a software-defined signal processing module capable of simultaneously processing signals from multiple objects and remotely communicating with a ground control system (GCS) for tracking and planning. In our system, the ArduPilotMega (APM) on the IRIS+ UAV transmits back its global positioning system (GPS) location to the Telemetry Host tool developed by our group to communicate with the APM module using the MAVLink protocol over a 915 MHz full duplex radio channel. The sensor system together with the antenna, SDR receiver, and the embedded compute module delivers objects' RSSI data through a 2.40 GHz radio channel to the GCS.

GPS locations of the UAV platform and objects' RSSI data are delivered to our tracking and planning algorithm—*solver*—through the Telemetry Host using a RESTful web service. The solver estimates object locations and calculates new control actions per each POMDP cycle to command the UAV through MAVLink to fly to a new location. In order to ensure safety and meet University regulatory requirements, we also employ QGroundControl—a popular cross-platform flight control and mission planning software—to monitor and abort autonomous navigation. We detail our sensor system below.

**Signal Processing Module:** Figure 3.3 illustrates the components of the proposed signal processing module. We propose using a *software-defined radio* (SDR) receiver to implement the signal processing components. The key advantages of our choice are the ability to: *i)* reduce the weight of the receiver; *ii)* rapidly scan a large frequency spectrum to track multiple animals beaconing on different VHF frequency channels;

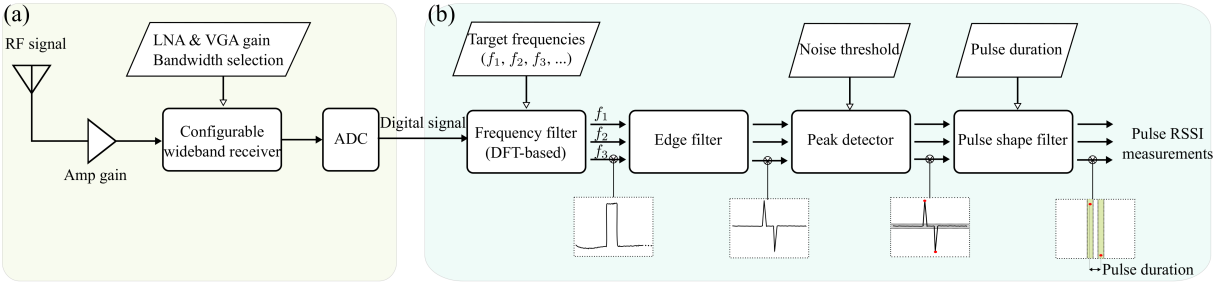


**Figure 3.3. The signal processing module.** (a) *Software-defined radio:* raw input RF signals are processed through the HackRF One SDR device with different configurable amplifiers–low noise amplifier (LNA) and variable gain amplifier (VGA), and an ADC to convert analogue signals to digital signals. (b) *Embedded compute module:* digital signals are processed on an Intel Edison board using a DFT (discrete Fourier transform)-based frequency filter with configurable input frequencies, edge filter and peak detector algorithms to derive radio collar RSSI measurements.

and *iii)* reconfigure the system on the fly because the signal processing chain is defined in software.

In this chapter, we use the *HackRF One* SDR—an open source platform developed by (Ossmann, 2015) capable of directly converting radio frequency (RF) signals to digital signals using an analogue-to-digital converter (ADC)—together with an Intel Edison board as our embedded compute module. We implemented a discrete Fourier transform (DFT) filter to isolate, from multiple signals, each unique VHF frequency channel associated with an animal radio collar and measure the signal strength of the received signal.

**Antenna:** A lightweight folded 2-element Yagi antenna was specially designed for our sensor system. Our design achieves a low profile antenna capable of being within the form factor of low-cost commodity UAVs suitable for easy operation in the field. Similar to a standard 2-element Yagi antenna, the folded design has one reflector and one driven element as shown in Figure 3.2b.

The antenna operates in the frequency range from 145 MHz to 155 MHz (a typical range for wildlife radio tags), and a centre frequency of $f_c = 150$ MHz. The length of driven and reflector elements are $D_d = 0.3975\lambda$ and $D_r = 0.402\lambda$, respectively, while $d_1 = 0.1\lambda$, $d_2 = 0.03\lambda$ and the inductive loading ring diameter is $d_3 = 0.015\lambda$. Here, the wavelength $\lambda = c/f_c = 2$ m with $c = 3 \cdot 10^8$ m/s. The antenna gain model calculated for the design is shown in Figure 3.6b.

### 3.4.1 Planning implementation for a real-time system

Implementing planning algorithms on any real-time systems is always challenging because of its high computational demand. Thus, in the following, we present the approaches to minimise the planning computational time while not sacrificing the overall localisation performance:

1. Notably, for RSSI data, the uncertainty in the estimation of an object's location is reduced when the maximum gain of the directional antenna mounted on the UAV points toward the object position. Hence, to increase the localisation accuracy, the UAV heading angle $\theta_k^u$ must be controlled during the path planning process, although the multi-rotor UAV can be manoeuvred without changing its heading. We select a set of discrete UAV rotation angles for the control actions $\mathcal{A}_k$ based on a simulation-based study to reduce the computational complexity of the POMDP planning process by limiting the number of possible actions to evaluate.

2. The solver performs planning in every $N_p$ observation cycles with $N_p > 1$ instead of every observation. This approach helps to ensure that the solver prioritises its limited computational resource on tracking objects instead of only performing planning steps.

3. A coarse planning interval $t_p$ in the planning procedure is implemented to minimise the computational time by reducing the number of look-ahead steps while still having the same look-ahead horizon. For example, if we want to estimate the object's state in a 10 second horizon, we can use the normal interval $t_p = 1$ s and estimate the object's state 10 times or use the coarser interval $t_p = 5$ s and perform the estimation twice; the latter approach is computationally less expensive.

4. Instead of selecting the best action from the possible action space $\mathcal{A}_k$, the domain knowledge of the receiver antenna gain is used to select a subset of actions that give the highest received gain using **Algorithm** 3.1.

Following the above implementation approach, UAV motion includes two modes: *i)* changing its heading angle while hovering, and *ii)* moving forward to its direct location. In one planning procedure with $N_p$ cycles, the UAV needs $\lfloor |\triangle\theta| / \theta_{max} \rfloor$ cycles

**Algorithm 3.1** Calculate the control action subset

1: **Input**: $N_{\mathcal{A},s}$, $\mathcal{A}_k$, $G_r$, $x_{k+H}$ ▷ action space, antenna gain and object's position
2: **Output**: $\mathcal{A}_k^s$ ▷ subset of control actions
3: **for** $l = 1 : N_{\mathcal{A},k}$ **do**
4: Get $u_{k+H}^l \in \mathcal{A}_k(l)$
5: Calculate $G_r^l := G_r(x_{k+H}, u_{k+H}^l)$
6: **end for**
7: $\mathcal{A}_k^s := \mathcal{A}_k(G_r^l \geq$ **Top** $N_{\mathcal{A},s}$ of $G_r)$ ▷ select top $N_{\mathcal{A},s}$ with highest gains

to rotate, and spends the remaining cycles $N_p - \lfloor |\triangle\theta| / \theta_{max} \rfloor$ to move forward without changing its heading. Here $\lfloor \cdot \rfloor$ and $| \cdot |$ are the floor and absolute operator respectively, and $\theta_{max}$ is the UAV maximum rotation angle in one cycle . The sign of $\triangle\theta$ decides the rotation direction ($+$ for the clockwise, and $-$ for the counter-clockwise).

## 3.5 Simulation Experiments

Implementing on a real system is time-consuming and difficult. Hence, we want to validate our systems first through several simulation experiments to: *i)* verify our tracking and planning algorithms; *ii)* investigate how our planning parameters such as different $\alpha$ values of the Rényi divergence or the number of discrete actions $N_{\mathcal{A},s} = |\mathcal{A}_k^s|$ created in Algorithm 3.1 contribute to the overall algorithm performance; and *iii)* compare our proposed Rényi divergence based planning technique with other well-known methods, and the impact of the look-ahead horizon parameters on computational time and localisation accuracy. All of the simulation experiments were processed on a PC with an Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz, 32GB RAM and MATLAB-2016b.

### 3.5.1 Tracking and planning simulation

This simulation was implemented to validate our approach under a synthetic scenario where all parameters (*e.g.*, velocity of the UAV $v_u$) are set to those expected in practice. In this experiment, the UAV attempted to search and localise 10 moving objects randomly located in an area of 500 m $\times$ 500 m. The following are the list of parameters used in this simulation: the sampling time step is 1 second since the tag emits pulse signals every 1 second. The solver performed a planning procedure every $N_p = 5$ s, and the look-ahead horizon parameters: $H = N_H t_p = 5$ s with the number of horizons

$N_H = 1$ and the planning interval $t_p = 5$ s. The UAV started from its home location at $u_1 = [0, 0, 20, 0]^T$ m, moved under the constant velocity $v_u = 5$ m/s with its maximum heading rotation angle $\theta_{max} = \pi/6$ rad/s. The number of particles for each object was capped at $N_s = 10,000$, with the number of future sample measurements $M_s = 50$, the Rényi divergence parameter $\alpha = 0.5$, the number of actions $N_{\mathcal{A},s} = 5$. In addition, an object is considered localised if its location uncertainty, determined by the determinant of its particles covariance, is small enough—$N_{Th} = 10,000$ m$^{2N_s}$ was chosen as the limit. The *LogPath* measurement model with $P_r^{d0} = 7.7$ dBm, $n = 3.1, \sigma_P = 4.22$ dB was used to verify our proposed algorithm. To demonstrate that our algorithm was able to localise mobile objects, a *wombat*—an animal that usually wanders around its area was considered. Hence, a *random walk* model was used to describe its behaviour with a single object's transitional density:

$$f_{k|k-1}(x_k|x_{k-1}) = \mathcal{N}(x_k; Fx_{k-1}, \mathcal{Q}), \tag{3.9}$$

where $F = I_3$ with $I_n$ is $n \times n$ identity matrix , $\mathcal{Q} = \sigma_Q^2 \,\mathrm{diag}([1, 1, 0]^T)$, $\sigma_Q = 2$ m/s.

Figure 3.4 shows localisation results for 10 mobile objects where the estimation details are annotated next to the object's position with two indicators: *Root Mean Square (RMS)* and *flight time*—see Section 3.5.2 for definitions. In summary, for this scenario, it took the UAV **587** seconds to localise all ten moving objects at a maximum error distance of less than 15 m, except for an outlier, object 2 (RMS = 26.3 m). At flight time 587 s, after localising the last object (object 7), the UAV was sent a command to fly back to its original home location. In this case, the total UAV travel distance was 1.93 km. The results demonstrate that our algorithm can search and accurately localise multiple numbers of objects in real time (about 10 minutes) and the travel distance 1.93 km is well within the capacity of commercial off the shelf drones under the 2 kg mass category.

### 3.5.2 Monte Carlo simulations

For this experiment, all of the Monte Carlo setup parameters were kept the same as in Section 3.5.1, except for those under investigation. In addition, to ensure that the results were not random, all of the conducted experiments were performed over 100 Monte Carlo runs. The tracking algorithm was evaluated based on the following criterion:

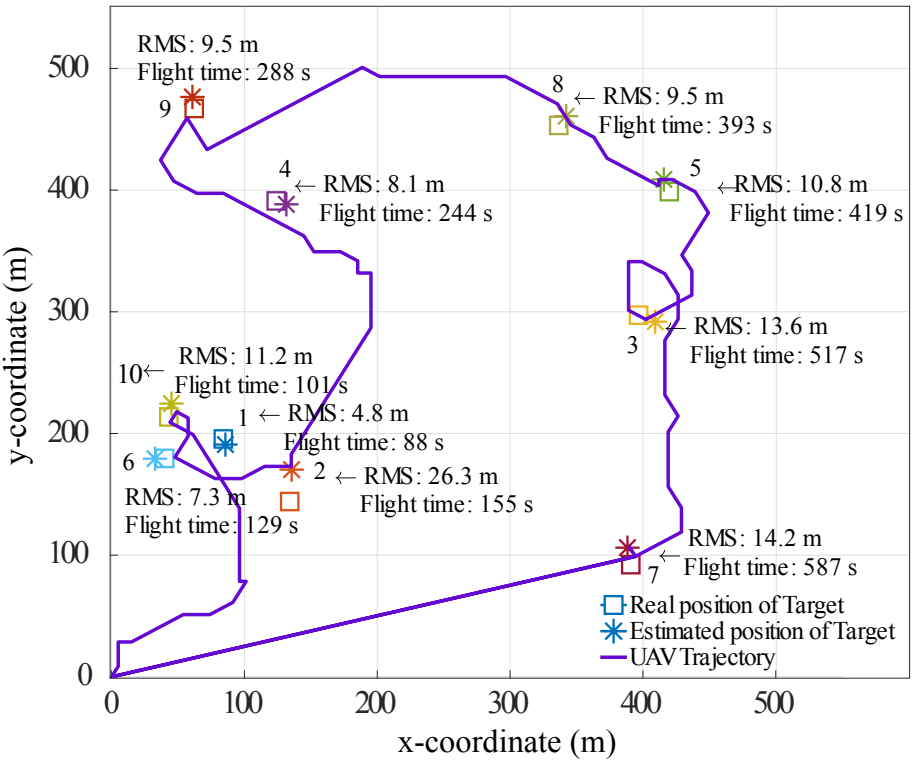


**Figure 3.4. Simulation results.** Ten mobile objects localised using a single UAV.

- *Estimation Error* is the absolute distance between ground truth and estimated object location $\mathcal{D}_{rms} = \sum_{j=1}^{N_{tg}} d^j_{rms}/N_{tg}$ with $d^j_{rms} = [(x^j_{truth} - x^j_{est})^2 + (y^j_{truth} - y^j_{est})^2]^{1/2}$;
- *Flight time* (s) for UAV to localise **all** of the objects and this includes hovering time when the UAV waits for commands from the solver to take an action;
- *UAV travel distance:* the total distance travelled by the UAV to track and locate all of the objects to the required location uncertainty bound; i.e the determinant of covariance being adequately small—$N_{Th} \leq 10,000\ \mathrm{m}^{2N_s}$;
- *Computational cost*: We evaluate the computational cost in terms of two components: *i)* execution time for the solver to execute the tracking algorithm only (called *non-planning time*), and *ii)* the execution time for the solver to select the best action—planning step—as well as complete the tracking task (called *planning time*).

First, our search and localisation algorithms were evaluated using different $\alpha$ values for the Rényi reward function in (3.7). Table 3.1 presents the Monte Carlo results for $\alpha = \{0.1, 0.5, 0.9999\}$. In general, the $\alpha$ value does not significantly impact the overall

**Table 3.1.** Localisation performance for different alpha values.

| | $\alpha = 0.1$ | $\alpha = 0.5$ | $\alpha = 0.9999$ |
|---|---|---|---|
| RMS (m) | **12.35** | 12.77 | 12.96 |
| Flight time (s) | **724** | 741 | 727 |
| UAV travel distance (km) | 2.38 | 2.41 | **2.34** |

performance. However, applying $\alpha = 0.1$ provides the best localisation results in terms of estimation error and search duration. Applying $\alpha = 0.5$ proposed in (Ristic and Vo, 2010; Ristic, Morelande and Gunatilaka, 2010) results in the worst performance, it increases flight time and travel distance necessary to complete the localisation task. Using $\alpha = 0.9999$ (considered as using KL divergence which is a popular information gain measure) helps to save UAV travel distance while sacrificing location accuracy. One explanation for this scenario is that our noisy measurement causes the posterior density $\pi_{k+H}(\cdot|z_{1:k}, z^{(m)}_{k+1:k+H}(a))$ in (3.7) to be less informative due to high uncertainty. Therefore, the reward function should place more emphasis on the current posterior instead by using a small $\alpha$ value or setting $\alpha \to 1$ to completely ignore the future posterior. This also explains the reason for the worst localisation performance observed when $\alpha = 0.5$ (equally weighting the current and the future posterior).

Second, we conducted experiments to understand how the number of actions $N_{\mathcal{A},s}$ created by Algorithm 3.1 affects our tracking performance in term of planning time and localisation error. Table 3.2 shows Monte Carlo results for $N_{\mathcal{A},s} = \{2, 3, 4, 5, 6, 7\}$ wherein the planning time increases linearly with respect to the number of actions. Further, increasing the number of actions beyond four does not necessarily lead to better planning decisions because of the directionality of the antenna gain. Since the antenna gain is not omnidirectional, some actions result in changing the heading where antenna gain along the propagation path between the UAV and the object is lower; when the number of actions evaluated is increased, we encounter instances when an action leading to such a lower antenna gain results in a higher reward. This result is a consequence of the inherent uncertainties in the models used in tracking and planning. Thus, $N_{\mathcal{A},s} = 4$ provides an adequate pool of actions to yield the best localisation performance in terms of estimation error, flight duration, and travel distance, a desirable result for realising real-time planning with limited computational resources.

**Table 3.2.** Localisation performance for different number of actions.

| Number of actions $N_{\mathcal{A},s}$ | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| RMS (m) | 14.18 | 12.64 | **12.17** | 12.27 | 12.83 | 12.63 |
| Flight time (s) | 840 | 781 | **693** | 723 | 756 | 799 |
| UAV travel distance (km) | 2.62 | 2.53 | **2.39** | 2.50 | 2.52 | 2.70 |
| Planning time (s) | **1.16** | 1.19 | 1.23 | 1.27 | 1.36 | 1.47 |

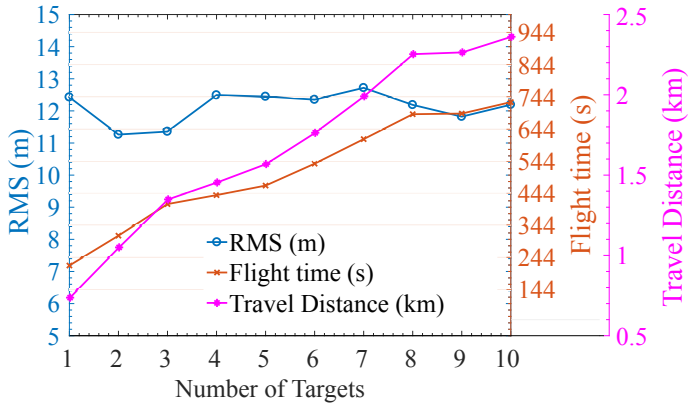


**Figure 3.5. Localisation performance for different number of objects.** $N_{tg}$ increases from 1 to 10.

Third, we want to examine the performance of our proposed algorithm under an increasing number of objects; in this study, we increase the maximum number of objects $N_{tg}$ from 1 to 10. As depicted in Figure 3.5, our algorithm's estimation error was stable and invariant to the number of objects. Moreover, it is reasonable that the flight time and the travel distance increased linearly with object numbers because it took more time and power to track more objects.

Fourth, we examined the performance of the information gain measure, Rényi divergence, under different look-ahead horizons $H = N_H t_p$ compared to: *i)* Shannon entropy (Cliff et al., 2015), *ii)* a naive approach that moves UAV to the closest estimated object location, and *iii)* a uniform search with the predefined path used in (Ristic, Morelande and Gunatilaka, 2010). Table 3.3 shows the Monte Carlo comparison results among various planning algorithms. All the parameters were reused from the Section 3.5.1, except for $\alpha = 0.1$ and $N_{\mathcal{A},s} = 4$ that was updated based on the previous experimental results. The results demonstrated that the Rényi divergence based reward function leads to significantly better planning strategies in comparison with other reward functions in terms of localisation accuracy, including Shannon entropy with the same horizon settings ($N_H = 1; t_p = 5$). For the Rényi reward function

**Table 3.3.** Localisation performance for different planning algorithms.

| | **Uniform** | **Closest object** | **Shannon** (Cliff et al., 2015) | **Rényi** | | | |
|---|---|---|---|---|---|---|---|
| $N_H$ | N/A | N/A | 1 | 1 | 3 | 5 | 10 |
| $t_p$ (s) | N/A | N/A | 5 | 5 | 1 | 1 | 1 |
| RMS (m) | 18.8 | 13.4 | 12.6 | 12.5 | 12.4 | 12.0 | **11.6** |
| Flight time (s) | 921 | 799 | 774 | **699** | 889 | 811 | 822 |
| UAV travel distance (km) | 3.72 | 2.29 | 2.54 | **2.27** | 2.99 | 2.82 | 2.42 |
| Planning Time (s) | 1.58 | **1.11** | 1.38 | 1.28 | 1.53 | 1.65 | 2.71 |
| Non-planning Time (s) | 1.58 | 1.03 | 0.99 | 0.97 | **0.96** | 0.97 | **0.96** |

itself, the large look ahead horizon number $N_H > 1$ helps to improve the localisation accuracy; however, it requires higher computational power (planning) and causes the UAV to travel further. Using $N_H = 1; t_p = 5$ s provides the best trade-off between computational time and accuracy.

**Summary:** According to the above simulation results, we select $\alpha = 0.1$, $N_{\mathcal{A},s} = 4$, and $N_H = 1, t_p = 5\ s$ as the planning parameters for the field experiment since these parameters provide the lowest computational cost, best performance in terms of location estimation error, travel distance and flight time.

## 3.6 Field Experiments

We describe here our extensive experiments regime to validate our approach and evaluate the performance of our aerial robot system in the field. Our aim is to: *i)* investigate the possibility of signal interference from spinning motors of a UAV on RSSI measurements; *ii)* estimate the model parameters in the sensor model and validate the proposed model; and *iii)* conduct field trials to demonstrate and evaluate our system capabilities.

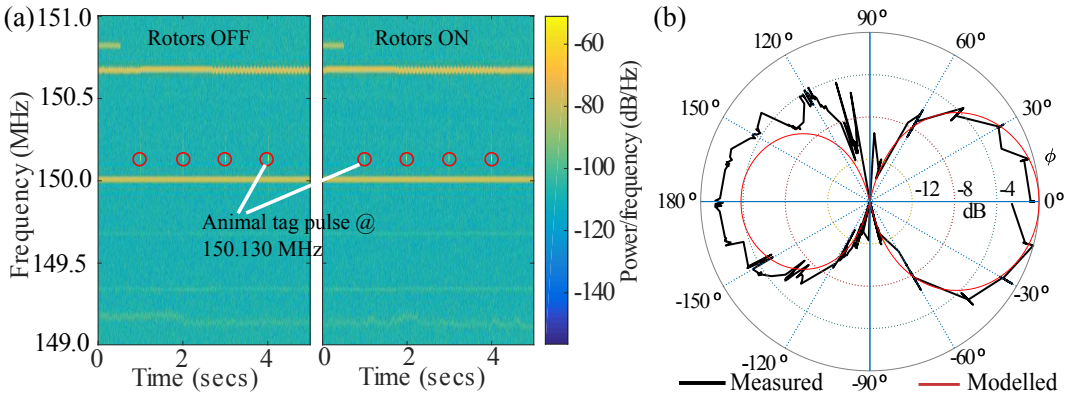


**Figure 3.6. Rotor noise and Antenna gain.** a) Waterfall plot for the rotor noise experiment when four motors spun at maximum rotation speed. b) Normalised antenna gain in E-plane $G(\phi)$. The red line is gain modelled pattern and black line is the normalised measured gain pattern from 30 measurements collected by rotating the UAV heading at 15° intervals.

### 3.6.1 Rotor noise

We investigated the rotor noise to confirm that our system is not affected by the electromagnetic interference from the UAV's motors. It also helps to clear the concern raised in (Cliff et al., 2015) that the rotor noise may affect the RSSI measurements. Four motors of the 3DR IRIS+ quad-copter shown in Figure 3.1 were used in this experiment. The RSSI data of a radio collar were measured across 149 MHz to 151 MHz frequency spectrum when four motors were operating at 20%, 50%, 100% of its maximum speed of 10,212 revolutions per minute. Figure 3.6a shows the frequency spectrum of the received signal. We can see that there was no difference in the frequency characteristics when the rotors were in ON and OFF states. This result confirms that the rotors do not spin fast enough to generate high-frequency interference to impact our RSSI measurements.

### 3.6.2 Sensor model validation and parameter estimation

**Antenna Gain:** The antenna gain pattern was measured to verify its directivity compared to the antenna gain model $G_r(x,u) = G_r(\phi)$ calculated—following (Orfanidis, 2002, pp.1252)—based on the physical design as discussed in Section 3.4. Figure 3.6b shows the measured and modelled radiation patterns $G_r(\phi)$ in the E-plane. In the measurement process, $\phi$ is evaluated as the angle between the UAV heading, changed through 0° to 360°, and the direction from its position to a fixed location of a VHF radio tag. The result shows that the front-to-back

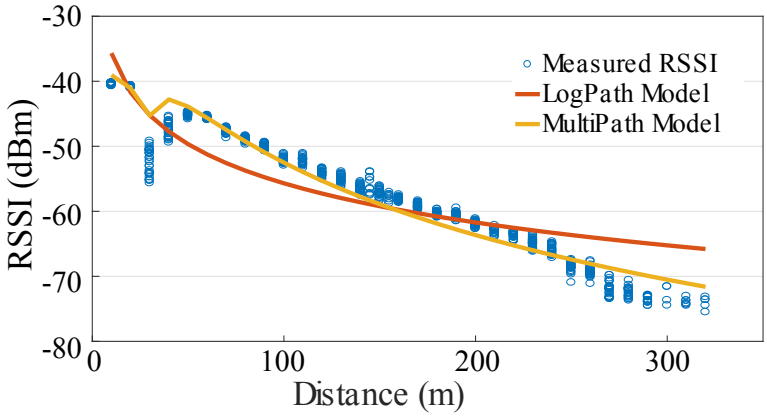


**Figure 3.7. Empirical measurement model.** Plot of measured RSSI data points and its model estimates over a distance from 10 m to 320 m at 10 m intervals.

ratio is smaller (2 dB) than expected and this is an artefact of folding the reflector on our design.

**Signal propagation model parameter validation:** We collected RSSI data points over a range from 10 m to 320 m between the UAV and a VHF radio tag. The tag and the UAV were kept at a height of 5 m above ground during this experiment. The tag was stationary at all times, while the UAV was directed to move away in a straight line from the tag at 10 m intervals whilst hovering at each location to allow the collection of approximately 30 measurements. The UAV heading was maintained to ensure consistent antenna gain during the experiment. Since we operated in an open terrain over a grassland, we selected the path loss exponent $n = 2$ suitable for modelling free space path loss. Figure 3.7 shows the measured RSSI and the propagation models obtained using a non-linear regression algorithm to estimate model parameters. We have the following results for reference power $P_r^{d0}$ in (3.3), (3.4) at the reference distance $d_0 = 1$ m, and measurement noise variance $\sigma_P$ in (3.5):

- **LogPath model :** $P_r^{d0} = -15.69$ dBm , $\sigma_P = 4.21$ dB,
- **MultiPath model :** $P_r^{d0} = -15.28$ dBm , $\sigma_P = 2.31$ dB.

The results show that both models, as expected, derived a similar reference power $P_r^{d0}$ whilst providing a reasonable fit to measurement data. This affirms the validity of our propagation model. Although the *LogPath* model is reasonable, the *MultiPath* model is more accurate and yields a smaller measurement noise variance. The results confirm the impact of ground reflections, especially close to the signal source.

### 3.6.3 Field trials

We designed and conducted two sets of field experiments that included 20 autonomous missions as described below.

- **First set of trials (autumn season):** We conducted a total of 16 autonomous flights with two mobile radio-tags to evaluate the measurement models and demonstrate the robustness of our system (see Section 3.6.4).

- **Second set of trials (winter and wet season):** We conducted 4 autonomous flights with the best performing measurement model. These experiments were aimed at demonstrating the multi-object tracking capability of our aerial robot platform under a mix of stationary (3 radio-tags) and mobile (2 radio-tags) object dynamics. In particular, we subjected our system to two highly mobile objects. Notably, these trials were conducted during the wet winter months when the test zone was over-grown with grass and shrubs. Therefore, these experiments demonstrate our system's capability to plan a trajectory to track multiple radio-tagged objects with differing motion dynamics and under different environmental settings (see Section 3.6.5).

Our experiments were designed around the University of Adelaide and CASA (Civil Aviation Safety Authority, Australia) regulations governing the conduct of UAV research. Given the need to operate in an autonomous mode, our flight zone, as well as the scope of the experiment, was restricted to University-owned property designated for UAV flight tests. Prior to gaining ethical and regulatory clearances to progress our field trial to a wildlife species of interest to conservation biologists, our first objective is to evaluate and demonstrate a robust working prototype. This is a necessary condition to gain both regulatory and ethical approval. Further, it is not feasible to have a wildlife species of interest at the remote test site and conduct experiments to systematically evaluate the aerial robot system. Therefore we chose to conduct experiments with human test subjects with stipulated safety measures in an area allocated for field tests. This allowed us to create various object motion dynamics as well as obtain accurate ground truth data for tag locations to evaluate our system. Notably, our measurement model is based on the received signal strength indicator (RSSI-based) measurements of signals transmitted from radio-tags. Hence, there is no technical difference whether the radio-tags are carried by humans or wildlife.

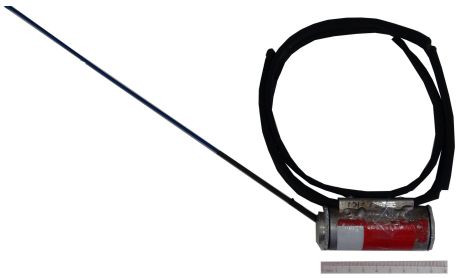

**Figure 3.8. Example of a radio-collared tag.** The collar used for radio tagging the endangered species of Southern Hairy Nose (SHN) wombats used in our field experiments. Each tag used in our experiments transmits an unmodulated on-off-keyed signal with a pulse width ranging from $10-20$ ms, at a period of approximately 1 s, and using a unique frequency in the range of $150-152$ MHz.

In the field trials, the task of the aerial robot system was set to search and localise radio-tags undergoing various motion dynamics in a search area 75 m $\times$ 300 m (2.25 hectares). Instead of wildlife, we relied on volunteers to wear a VHF radio tag of the type shown in Figure 3.8 on their forearm, and carry a mobile phone-based GPS data logger in their hands to obtain ground truth data. We were required to have two extra personnel stationed to maintain constant sight of the UAV as well a pilot with an RePL (remote pilot licence) in the field capable of aborting the autonomous mode and transferring control to manual operations mode.

### 3.6.4 First set of trials

In this section, we present the first set of field trials to demonstrate the planning method for tracking mobile objects. We also compare localisation performance between the two signal propagation models: *LogPath* model and *MultiPath* model derived in section 3.6.2. We used two VHF radio collars for these trials.

Figure 3.9 shows the tracking and localisation results along with UAV trajectories based on the two different measurement models. As expected, we observe the UAV planning has a tendency to approach the object's position since when the distance between the UAV and objects reduces, the RSSI measurement uncertainty is reduced. Thus it helps to reduce the uncertainty and increase the information gain. We can observe a clear difference in the *LogPath* model and *MultiPath* model where UAV pursues the second object after completing the tracking task for object 1. The more accurate *MultiPath* model is able to track and localise the second object without needing a close approach. We can also observe that using the *LogPath* model, where

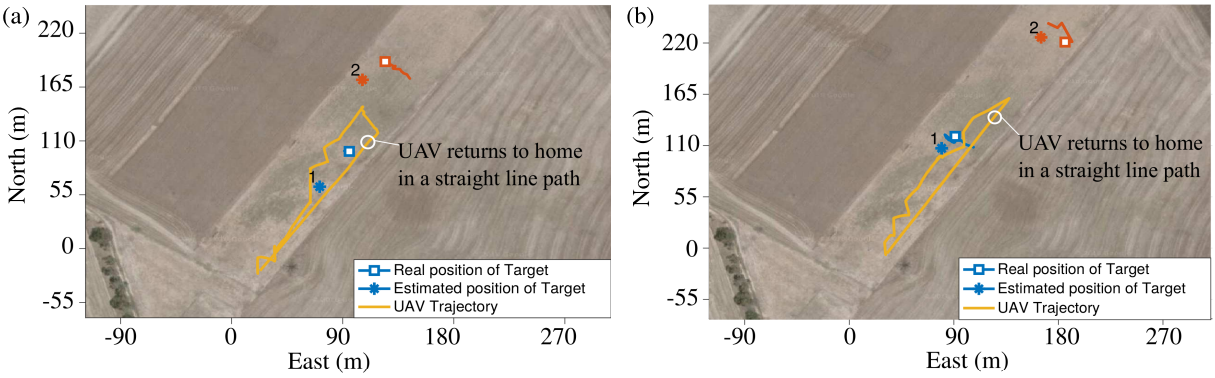


**Figure 3.9. Field experiment results.** Searching, tracking and localisation results of two mobile tags for the two different measurement models. a) Standard **LogPath**. b) **MultiPath**.

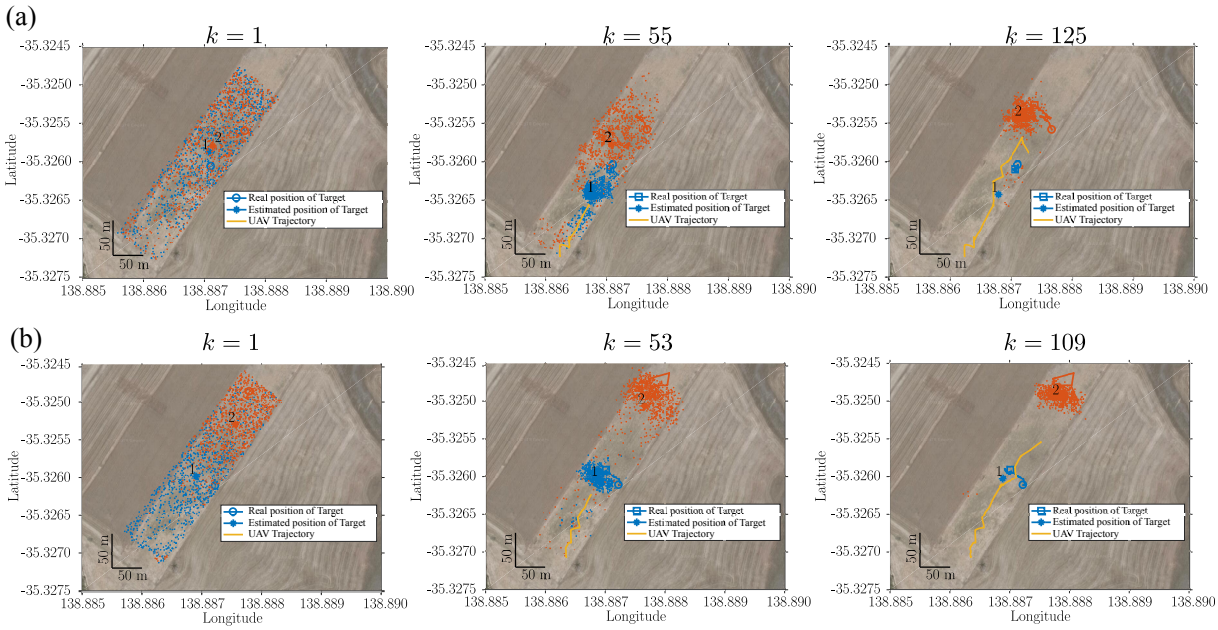


**Figure 3.10. The intermediate distributions of belief density representing the location of the radio-tags for the two scenarios in Figure 3.9.** Here, Figure 3.10a demonstrates the convergence of the belief density of the radio-tag positions using the standard **LogPath** measurement model in Figure 3.9a) after first observation ($k = 1$), tag 1 is localised ($k = 55$), and tag 2 is localised ($k = 125$). Similarly, Figure 3.10b demonstrates the convergence of the belief density of the radio-tag positions using the **MultiPath** measurement model in Figure 3.9b) after the first observation ($k = 1$), tag 1 is localised ($k = 53$), and tag 2 is localised ($k = 109$). The blue and orange dots represent the starting positions of tag 1 and tag 2, respectively. The square symbols denote the ground truths of the localised tags; the star symbols denote the estimated positions of the tags. The solid yellow lines represent the UAV trajectories.

multipath propagation is not modelled but is clearly dominant close to the object, leads to a poorer localisation accuracy despite the path planning algorithm leading the UAV close to the object.

Figure 3.10 shows the particle distribution after the first observation is updated and when the objects are tracked and localised using the two measurement models. We can see that the solver is able to estimate the two tag positions quite accurately even after the update using the initial observation; however, the uncertainty (as noted by the particle distribution) is still very high. Interestingly, *MultiPath* model location uncertainty is significantly less where object 1 is placed in the bottom half of the field while object 2 is placed in the top half of the field. Object 1, being closer to the UAV, is localised first, with under 55 measurements for both measurement models. At the time when object 1 is localised, the uncertainty of object 2 is relatively higher for the *LogPath* model. The *MultiPath* model required significantly fewer measurements to track and localise object 2. As expected, both measurement models required significantly more measurements to localise the second object given the high measurement uncertainty associated with being much further than the first object from the UAV during its flight. Furthermore, the random walk of the second object provided a challenging scenario since object 2 typically moved a larger distance around the field compared to the random walk performed by object 1.

Although the solver guides the UAV to move toward an object's position in both measurement models, as expected, the standard *LogPath* model is less accurate compared to the *MultiPath* model shown in Figure 3.7; thus, the uncertainty when using the *LogPath* model is higher and leads to longer time duration to localise the two tags. Albeit model uncertainty, the *LogPath* model is still capable of locating both moving objects within the flight time capability of the UAV. The consequence of model uncertainty resulting from the *LogPath* model is more apparent when the UAV makes an approach to the object and the distance to the object is less than 50 m. This is evident in comparing the belief density in Figure 3.10a at $k = 125$ to that in Figure 3.10b at $k = 109$. We can see that the object location uncertainty increases for the *LogPath* model in the vicinity of 50 m and as a result, the UAV requires an increased number of manoeuvres to track and locate the object.

Table 3.4 presents the summary comparison results of location estimates between the two measurement models. Smaller RMS (root mean square) estimation error values suggest a higher accuracy, while shorter flight times and travel distance to

**Table 3.4.** Comparison of localisation performance.

| Model | Object Type | Trials | RMS (m) | Total Flight Time (s) | Travel Distance (m) |
|---|---|---|---|---|---|
| LogPath | Mobile | 8 | $30.1 \pm 12.8$ | $255 \pm 104$ | $549 \pm 167$ |
| MultiPath | Mobile | 8 | $22.7 \pm 13.9$ | $138 \pm 53$ | $286 \pm 121$ |
| (Cliff et al., 2015) | Stationary | 6 | $23.8 \pm 14.0$ | 838[3] | N/A |

localise all objects are highly desirable for a practicable system given the power constrained nature of commodity UAVs. The results confirm that the *MultiPath* model is superior to the standard *LogPath* model since it has been able to account for ground reflections. Further, the UAV is not required to approach the object closely to reduce its measurement uncertainty when using the *MultiPath* model.

The results in Table 3.4 also demonstrate that our proposed method can localise two *mobile* objects with a shorter flight time and better accuracy compared to the method in (Cliff et al., 2015). The RMS flight time realised with the *MultiPath* model is one-sixth of that in (Cliff et al., 2015). Although our experiments were not performed with a live object animal species of interest to conservation biologists, we search and locate two mobile radio-tagged objects. In contrast, the (Cliff et al., 2015) method was formulated and implemented to locate a single stationary object. However, the approach in (Cliff et al., 2015) was evaluated with a stationary radio-collared live bird while our field experiments were conducted with human test subjects.

### 3.6.5 Second set of trials

In this section, we present the second set of field trials. We use the *Multipath* measurement model because it provides a better measurement likelihood as shown in the tracking accuracy and flight time results in Table 3.4. We can see from Figure 3.3, the SDR-based signal processing architecture used in our system scales to enable tracking a large number of radio tags. The number of VHF radio-tags that can be tracked and localised is only limited by the hardware, such as the battery life of the UAV and the receiver noise of the SDR. In order to demonstrate scalability and robustness,

[3]Information regarding the total flight is not reported in (Cliff et al., 2015), however, as shown in Figure 9 in (Cliff et al., 2015), one observation took 76.21s and one trial needed 11 observations, hence total flight time is $11 \times 76.21 = 838.31s$

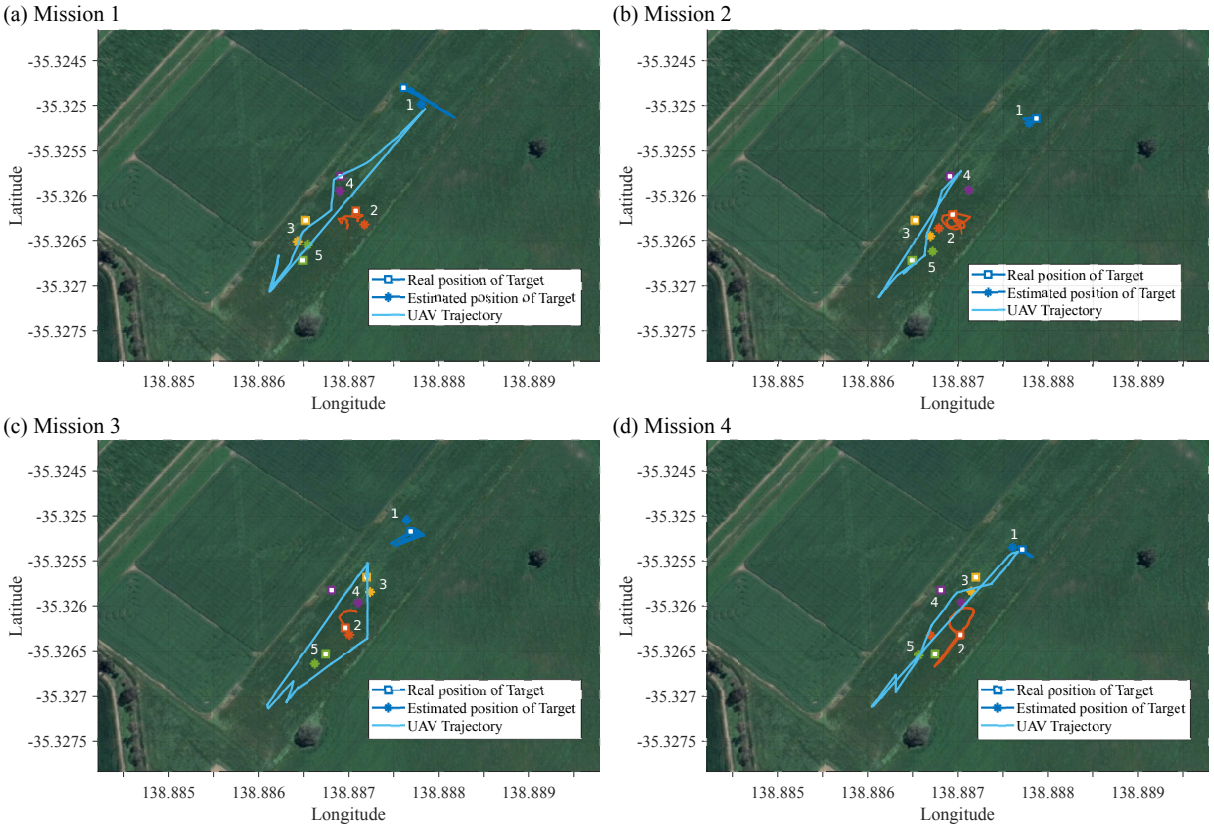


**Figure 3.11. Four autonomous field experiment missions.** Search, track and localise five objects: two mobile objects (object 1, 2) and three stationary tags (object 3, 4, 5). Figure 3.11(a), (b), (c), and (d) corresponds to the sequence of the missions in Table 3.5. The square symbols denote the ground truth of the localised radio-tags; the star symbols denote the estimated positions of the radio-tags; the solid blue lines represent the trajectories planned by the autonomous aerial robot to track the set of five VHF radio-collared tags.

we used five radio-tags. In order to demonstrate the capability of our system to accommodate different animal behaviours, we used two highly mobile objects (object 1, 2) and three stationary objects (object 3, 4, 5). Further, to demonstrate the robustness of our measurement model, we conducted these trials in the wet, winter season in South Australia where the test site was representative of a grassland with shrubs and moisture. We conducted four field missions in which the task of our aerial robot system was to track and localise five objects as opposed to two mobile objects investigated in Section 3.6.4. All other experimental settings were as described in Section 3.6.4.

Figure 3.11 depicts the UAV and mobile object trajectories together with tracking and localisation results. Table 3.5 presents a quantitative summary of the results from the four field missions. The results show that when the objects are highly mobile, such

**Table 3.5.** Localisation performance over for field missions to track and localise five radio-tagged objects.

| | RMS (m) | | | | | | Flight time (s) |
|---|---|---|---|---|---|---|---|
| **Object dynamics** | *Mobile* | | *Stationary* | | | **Mean** | |
| **Object #** | **Object 1** | **Object 2** | **Object 3** | **Object 4** | **Object 5** | | |
| **Mission 1** | 27.3 | 19.1 | 27.2 | 18.1 | 19.9 | 22.3 | 163 |
| **Mission 2** | 9.3 | 21.8 | 24.9 | 25.4 | 23.7 | 21.0 | 143 |
| **Mission 3** | 15.0 | 9.3 | 18.7 | 30.6 | 16.3 | 18.0 | 128 |
| **Mission 4** | 10.0 | 29.6 | 18.4 | 25.1 | 16.6 | 19.9 | 165 |

as object 1 in Figure 3.11a or object 1 and 2 in Figure 3.11d, the UAV takes longer flight paths to be able to localise these highly mobile objects. This is because the UAV undertakes control actions to position itself to reduce measurement uncertainty. Consequently, we also see that the UAV path planning algorithm undertakes control actions to navigate the UAV closer or follow objects to quickly reduce measurement uncertainty. In contrast, when the objects are less mobile as shown in Figure 3.11bc, the UAV can easily localise the objects with fewer measurements, shorter flight paths, and without needing to approach the objects. Thus, when objects are less mobile, the UAV requires less flight time to accurately track and localise them. We can see that our planning for tracking approach was robust with respect to various object motion dynamics we have created. Further, the results summarised in Table 3.5 demonstrate that our localisation results were consistently high across all four missions.

As expected, our aerial robot system can successfully track and localise multiple radio tags. In relation to the first set of field trials, we can also see that our system is: *i)* scalable to a larger number of VHF radio-tags, *ii)* robust against variations in environmental conditions, and *iii)* robust with respect to various object behaviours.

## 3.7 Discussion

In this section, we summarise and discuss results from our approach as well as compare and discuss our results in the context of the recent study by (Cliff et al., 2015) (see Section 3.7.1). We then reflect upon the lessons learnt from our field trials to build, test and evaluate a new approach following a different school of thought for autonomous tracking and localisation of VHF radio-tags (see Section 3.7.2). Our work, being a first, is not without limitations. We discuss these in Section 3.7.3.

### 3.7.1 Comparison

Table 3.6 presents a complete comparison between our proposed system and (Cliff et al., 2015) system. Notably, our search area is smaller compared to (Cliff et al., 2015) (75 m $\times$ 300 m v.s 1000 m $\times$ 1000 m) due to our test flight zone restrictions; however, we have set up our initial distance from the UAV home position to its farthest object's position to be equivalent to the distance of the stationary object in (Cliff et al., 2015), approximately 300 m. Although we have tried to replicate the distance to the location of a radio-tag, the detection range is determined by a number of factors other than the specification of the receiver and the antenna used. The detection range is heavily influenced by the transmitted power of a radio-tag, which is adjusted based on application requirements and varies in different environments, even for the radio collars from the same manufacturer. Therefore, we have not directly compared the detection range. Instead, we have tried to achieve a similar UAV-to-object distance in our experimental settings.

In general, as shown in Table 3.6, our system is more compact, lighter, and has a payload that is one-third of that in (Cliff et al., 2015) and consequently capable of longer flight times on any given UAV. Our total system mass being under 2 kg enables ecologists in jurisdictions such as Australia (Civil Aviation Safety Authority, 2017), Germany (Federal Ministry of Transport and Digital Infrastructure, 2017) and India (Office Of The Director General Of Civil Aviation, 2018) to operate our system without a remote pilot licence (RePL) and regulatory burdens. Moreover, as shown in Table 3.4, compared to the bearing-only method requiring full rotations of a UAV at each observation point, the ability to instantly collect RSSI measurements also helps reduce flight times significantly. Furthermore, as discussed in (Arulampalam et al., 2002), the computational cost for grid-based methods used in (Cliff et al., 2015) increases dramatically with the number of cells whilst the grid must be dense enough to achieve accurate estimations; *e.g.*, a grid-based filter with $N$ cells conducts $\mathcal{O}(N^2)$ operations per iteration, while a similar particle filter with $N$ particles only requires $\mathcal{O}(N)$ operations. Hence, the grid-based filter method is only suitable for cases with stationary objects as in (Cliff et al., 2015) where the most expensive computational step, the prediction step, is skipped. Moreover, as shown in Table 3.3, our planning algorithm based on Rényi divergence is superior to the Shannon entropy approach in (Cliff et al., 2015) in terms of two important metrics: *i)* accuracy, and *ii)* UAV flight time.

**Table 3.6.** Comparison between our system and (Cliff et al., 2015) system.

| | Ours | (Cliff et al., 2015) |
|---|---|---|
| *Payload* (g) | 260 | 750 |
| *Total mass (g)* | 1,280 | 2,200 |
| *Drone type* | Quad-copters (smaller drone) | Octocopters (relatively larger drone) |
| *Receiver Architecture* | Software defined radio (digital-based, rapidly scan multiple frequencies to support detecting signals from multiple animals) | Analogue filtering circuit and a fixed frequency narrowband receiver (analogue-based, difficult to re-configure for a new frequency) |
| *Antenna elements* | Compact, lightweight, folded 2-element Yagi antenna (designed for small drone form factor) | Antenna array structure requiring a large spatial separation of two antenna elements and wire ground plane |
| *Measurement model* | Range-only (exploiting the simplicity of a range-only measurement system) | Bearing-only (antenna array, and UAV rotation at grid points with a phase difference measurement system) |
| *Filtering method* | Particle filter ($\mathcal{O}(N)$ operations per iteration) | Grid-based filter ($\mathcal{O}(N^2)$ operations per iteration) |
| *Planning algorithm (reward function)* | Rényi divergence | Shannon entropy |
| *Objects dynamics* | Multiple mobile objects | A single stationary object |
| *Nature of objects* | Radio tags carried by humans test subjects | A radio-tagged bird (Manorina Melanocephala) |

The studies in (Dos Santos et al., 2014) and (VonEhr et al., 2016) also used an SDR receiver and considered the problem of detecting multiple VHF radio-tag signals using a software defined radio based receiver. We can make the following observations regarding the other SDR based receiver approaches:

- The team in (Dos Santos et al., 2014) used an SDR payload on a UAV flying a predefined flight path to store raw signal detection. This data was post-post processed after the flight to build a signal heat map. The detection range reported in (Dos Santos et al., 2014) is 240 m, similar to our range of 320 m.

- This study in (VonEhr et al., 2016) discussed two software defined radio methods to collect VHF signal measurements: *i)* using the Doppler effect, *ii)* bearing

measurements obtained by rotating a drone-mounted Yagi antenna, the so-called Yagi rotation methodology. Notably, this measurement approach is like that proposed in (Cliff et al., 2015). Only the Yagi rotation methodology was implemented with a reported bearing measurement accuracy of $\pm 30$ degrees. More significantly, the detection range reported in (VonEhr et al., 2016) is up to 1.5 km. This is mainly due to a higher gain antenna (3-element Yagi vs 2-element Yagi of our system) and a more sensitive SDR, the Funcube Dongle Pro+ (FDP+) SDR used in the study. Although the Funcube Dongle Pro+ (FDP+) has a higher receiver sensitivity, it has a limited bandwidth compared to the HackRF One SDR device we employed.

The mass of the sensor systems was not reported in (VonEhr et al., 2016), but Funcube Dongle Pro+ (FDP+) SDR device with a mass of 17 g is significantly more lightweight than the HackRF One we employed with a mass of 100 g. Although detection range cannot be directly compared, we can see that together with a higher gain antenna, the hardware employed in (VonEhr et al., 2016) achieved a significantly larger signal detection range compared with our study and the studies in (Dos Santos et al., 2014) and (Cliff et al., 2015).

### 3.7.2 Lessons learnt

In this section, we share our observations and discuss lessons learnt during our extensive set of field experiments. We also share with the research community guidelines for establishing a framework for UAV operations and related research.

We realise that the field trials are difficult for any robotics system, especially for aerial platforms where several strict regulations govern their operation. These regulations can depend on jurisdictions under which the flight operations are conducted. Typically, regulations imposed can be different depending on the purpose of the flight such as commercial or recreational and the weight class of the UAV. Currently, there is a lack of harmonisation in these regulations. For instance, the requirement for a remote pilot licence (RePL) applies to countries such as Australia, Germany, and India only for UAVs over 2 kg (Civil Aviation Safety Authority, 2017; Federal Ministry of Transport and Digital Infrastructure, 2017; Office Of The Director General Of Civil Aviation, 2018). In contrast, New Zealand and Finland only require a licence for UAVs over 25 kg (Civil Aviation Authority Of New Zealand, 2015; Finnish Transport Safety

Agency, 2016). Therefore, the research team must *first* familiarise themselves with existing regulations governing the operation of UAVs. Second, the research team needs to negotiate with the insuring body under which they operate to allow the conduct of drone-based flights as this should not be assumed. Insurance agencies can place further restrictions upon the possible field trials that can be conducted due to legal and risk issues. Dealing with these critical issues first will allow getting a framework under which to operate UAV related research such as our work in this article. At the time of doing this research, such a framework was pioneered at our University. This included the creation of a chief remote pilot position and a maintenance controller position, and subsequently, applying to CASA (Civil Aviation Safety Authority, Australia) to obtain a remotely piloted aircraft operator's certificate (ReOC) to conduct UAV missions. The chief remote pilot registered with CASA then has the authority to evaluate, manage and approve all UAV flights conducted by University staff and students.

We observed, in both field experiments and simulations, that flying the robot platform higher allows obtaining a better signal compared to ground-based systems. This is because the signal propagating to the UAV system entering an open airspace will be less attenuated than a signal propagating to a ground-based antenna and receiver system. This is because a signal propagating to a ground-based receiver will be more attenuated from potentially multiple radio wave scatters, reflectors, absorbers such as shrubs and grass in the intervening paths. Therefore, flying the robot at a higher altitude can increase the detection range. Notably, in practice, this height advantage is sometimes obtained by using lightweight aircraft and this is an expensive proposition.

The detection range of our current system is not comparable to handheld systems. However, we can see that to develop a mature tool that can function independently and survey a large area of land, we need a longer signal detection range. One simple approach to increase the range is to employ a preamplifier stage for the SDR we have used. An alternative approach is to consider an SDR device with greater sensitivity in the VHF band. For example, an earlier SDR based design (VonEhr et al., 2016) has achieved a 1.5 km detection range. Although we could not have benefited from such a long range given the limited University allocated space for testing, the study in (VonEhr et al., 2016) shows that a different SDR device based receiver can offer much longer detection range. Most notably, the SDR used in (VonEhr et al., 2016) with a mass of only 17 g can be used to replace the SDR of mass 100 g we have employed to realise a further reduction in the mass of the sensor system.

The current flight time for 3DR IRIS+ quad-copter carrying our sensor system is only around 10 minutes while the detection range of the type of VHF collar we have used is around 320 m. Thus, surveying a larger area in the order of several hundred hectares is not yet feasible for our battery-equipped UAV. However, assuming we employ the SDR receiver used in (VonEhr et al., 2016), we can achieve a reported detection range of 1.5 km. Consequently, we can see that such a detection range can achieve a survey area defined by a radius of 1.5 km to yield an area of over 700 hectares. Alternatively, if we assume that the survey area scales with the square of the detection range, we can see that an area of 225 hectares can potentially be surveyed.

Further, we observe that flying the UAV close to highly mobile objects helps to reduce localisation uncertainty. We can clearly observe this in our path planning results in Figure 3.11a where object 1 was running back and forth compared with the UAV trajectory for Figure 3.11b where object 1 was less mobile. However, a close approach by a UAV may disturb the wildlife of interest (Hodgson and Koh, 2016; Mulero-Pázmány et al., 2017) and can be potentially counterproductive when attempting to obtain accurate spatial and temporal information of threatened species. Wildlife reactions to a UAV differ among different species. For example, terrestrial mammals are less reactive to a UAV than birds (Mulero-Pázmány et al., 2017). Therefore, the potential for disturbance as well as operating parameters of a UAV close to wildlife is more likely to be dependent on the species of interest. We hope to be able to address questions around appropriate operating parameters for drones in our future work. Nevertheless, we should consider maintaining a safe distance from wildlife. A practical solution can be found by flying at the highest altitude possible (Mulero-Pázmány et al., 2017). A second approach is to use a receiver with a higher sensitivity, such as the hardware used in (Mulero-Pázmány et al., 2017), to increase the signal detection range. A third approach can be to reformulate the trajectory planning algorithm using the void probability functional proposed in (Beard et al., 2017). Such a planning method can alter the control decisions of the path planning algorithm to avoid approaching wildlife and always maintain a safe distance.

### 3.7.3 Limitations

While we have demonstrated a successful system, our approach is not without limitations.

Although we formulated a three-dimensional (3D) tracking problem—see equation (3.9)—our implementation assumed a fixed UAV altitude during the field trials. Therefore the implemented algorithm solved a two-dimensional (2D) tracking problem, that is ideally suitable for tracking and locating endangered species in largely flat terrains and grasslands. Consequently, the current approach is not suitable for tracking wildlife in hills or mountainous areas. See Appendix B for our initial investigation of tracking objects in 3D environments.

## 3.8 Conclusion

We have developed and demonstrated an autonomous aerial vehicle system for tracking and localising VHF radio-tagged animals using noisy RSSI based measurements and considered the mobility of objects during their discovery in the field. The joint particle filter and POMDP with Rényi divergence based reward function provided an accurate method to track and locate multiple animal collars while considering the resource constraints of the underlying UAV platform. In addition, we have realised a lightweight sensor system to minimise the payload on a UAV and achieved longer flight times.

Our problem formulation assumes that at least one object is visible or the UAV's initial heading can be in the general direction of the objects or the sensor has a very long detection range. This approach is similar to that followed in (Cliff et al., 2015). In future work, planning formulation should consider both exploration and tracking to deal with events where there are no detectable radio signals within the range of the sensor (Charrow, Michael and Kumar, 2015). See Chapter 5 for how we propose a new multi-objective algorithm to perform both searching and tracking of unknown number of objects.

Further, this chapter only considers the scenarios in high signal-to-noise ratio (SNR) environments where the UAV can easily detect objects using common signal thresholding methods, which can be categorised under detection-then-track (DTT) techniques. However, in the cases of noisy and low SNR conditions, DTT methods often fail to detect the objects of interest if the threshold value is high, or generate false-alarms if the threshold value too low. Therefore, in the next chapter, we derived a separable track-before-detect (TBD) likelihood function for radio-tagged

signals resulting in an efficient TBD filter that can detect and track an unknown and time-varying number of objects under a random finite set (RFS) framework.

# Chapter 4

# Planning for Detecting and Tracking Multiple Radio-tagged Objects

WE consider the problem of online path planning for joint detection and tracking of multiple unknown radio-tagged objects under noisy and low SNR environments and propose an online path planning algorithm with joint detection and tracking. A POMDP with a RFS TBD multi-object filter is derived, which also maintains a safe distance between the UAV and the objects of interest using a void probability constraint. In practice, the multi-object likelihood function of raw signals received by the UAV in the time-frequency domain is shown to be separable and results in a numerically efficient multi-object TBD filter. A TBD filter is developed with a jump Markov system to accommodate manoeuvring objects capable of switching between different dynamic modes. Our evaluations demonstrate the capability of the proposed approach to handle multiple radio-tagged objects subject to birth, death, and motion modes. Moreover, this online planning method with the TBD-based filter outperforms its detection-based counterparts, such as the method in Chapter 3, in tracking, especially in low signal-to-noise ratio environments.

## 4.1 Motivation and Contribution

In this chapter, we consider the problem of online path planning for UAV based localisation or tracking of a time-varying number of radio-tagged objects. This is an important basic problem if UAVs are to be able to autonomously gather spatial-temporal information about the objects of interest such as animals in wildlife monitoring (Kays et al., 2011; Thomas, Holland and Minot, 2012; Cliff et al., 2015; Nguyen et al., 2019a), or safety beacons in search-and-rescue missions (Gerasenko et al., 2001; Murphy et al., 2008). Signals received by the UAV's on-board radio receiver are used for the detection and tracking of multiple objects in the region of interest. However, the radio receiver has a limited range, hence, the UAV—with limited energy supply—needs to move within range of the mobile objects that are scattered throughout the region. This is extremely challenging because neither the exact number nor locations of the objects of interest are available to the UAV.

Detecting and tracking an unknown and time-varying number of moving objects in low signal-to-noise ratio (SNR) environments is a challenging problem in itself. Objects of interest such as wildlife and people tend to switch between various modes of movements in an unpredictable manner. Constraints on the transmitters such as cost and battery life mean that signals emitted from radio-tagged objects have very low power, and become unreliable due to receiver noise, even when they are within receiving range. The traditional approach of detection before tracking incurs information loss, and is not feasible in such low SNR environments. Reducing information loss introduces far too many false-alarms, while reducing the false-alarms increases misdetections and information loss (Lehmann, 2012).

Planning the path for a UAV to effectively detect and track multiple objects in such environments poses additional challenges. Path planning techniques for tracking a single object are not applicable. Since there are multiple moving objects appearing and disappearing in the region, following only certain objects to localise them accurately means that the UAV is likely to miss many other objects. The important question is: *which objects should the UAV follow, and for how long before switching to follow other objects or to search for new objects?* In addition to detection and tracking, the UAV needs to maintain a safe distance from the objects without exact knowledge of their locations. For example, in wildlife monitoring, UAV noise would startle animals away if they move within a close range. We also need to keep in mind that the UAV itself has limited power supply as well as computing and communication resources.

## 4.2 Related Work

Well-known bio-inspired planning algorithms such as genetic algorithm (GA) and particle swarm optimisation (PSO) (Roberge, Tarbouchi and Labonte, 2013) are computationally expensive and not suitable for online applications. Markov decision process and partially observable Markov decision process (POMDP) are receiving increasing attention as online planning algorithms over the last few decades with techniques such as grid-based MDP (Baek et al., 2013), or POMDP with nominal belief state optimisation (Ragi and Chong, 2013). Furthermore, at a conceptual level, the POMDP framework enables direct generalisation to multiple objects via the use of random finite set (RFS) models (Mahler, 2007b). Random finite set can be regarded as a special case of point process when the points are not repeated (for more information on point process theory, please see (Moller and Waagepetersen, 2003; Vo, Singh and Doucet, 2005; Daley and Vere-Jones, 2007)). This so-called RFS-POMDP is a POMDP with the information state being the filtering density of the RFS of objects.

RFS-POMDP provides a natural framework that addresses all the challenges of our online UAV path planning problem. Indeed, RFS-POMDP for multi-object tracking with various information theoretic reward functions and task-based reward functions have been proposed in (Ristic and Vo, 2010; Ristic, Vo and Clark, 2011; Hoang and Vo, 2014; Hoang et al., 2015; Beard et al., 2017) and (Gostar, Hoseinnezhad and Bab-Hadiashar, 2013; Gostar et al., 2017; Wang et al., 2018), respectively. This framework accommodates path planning for tracking an unknown and time-varying number of objects in a conceptually intuitive manner. In addition, RFS constructs such as the void probabilities facilitate the incorporation of a safe distance between the UAV and objects (whose exact locations are unknown) into the POMDP (Beard et al., 2017). However, these algorithms require detection to be performed before tracking and hence are not applicable to our problem due to the low SNR.

In our earlier work (Nguyen et al., 2019a) (see Chapter 3), we presented a path planning solution for tracking one object at a time, in a high SNR environment with a fixed number of objects. This solution, also based on a detection before tracking formulation, is not applicable to the far more challenging problem of simultaneously tracking an unknown and time-varying number of objects in low SNR.

In this chapter, we propose an online path planning algorithm for joint detection and tracking of multiple objects directly from the received radio signal in low SNR

environments. This is accomplished by formulating it as a POMDP with an RFS-based track-before-detect (TBD) multi-object filter.

TBD methods operate on raw, unthresholded data (Ebenezer and Papandreou-Suppappola, 2016) and are well-suited for tracking in low SNR environments such as infrared, optical (Barniv, 1985; Tonissen and Bar-Shalom, 1998; Rutten, Gordon and Maskell, 2005; Vo et al., 2010), and radar (Buzzi, Lops and Venturino, 2005; Buzzi et al., 2008; Lehmann, 2012; Dunne and Kirubarajan, 2013; Papi et al., 2015). However, TBD methods are computationally intensive, and TBD for range-only (received signal strength) tracking has not been developed. One of the main innovations of our solution is to convert the raw signals received by the UAV receiver into time-frequency input measurements for the multi-object TBD filter (using the short time Fourier transform). Such signal representation enables us to derive a separable measurement likelihood function that yields a numerically efficient multi-object TBD filter. In order to accommodate the time-varying modes of movements of the objects, we use a jump Markov system (JMS) to model their dynamics. Further, to maintain a safe distance from the objects, we impose an object avoidance constraint based on the void probability functional in (Beard et al., 2017) for the planning formulation.

## 4.3 Problem Formulation

### 4.3.1 Problem statement

The sensor system under consideration consists of a UAV with antenna elements, and a signal processing module. Following the sensor hardware description in Chapter 3, we present some of its basic components:

- UAVs used are commercial, civilian, low cost, and small form factor platforms with physical constraints on maximum linear and rotation speeds and onboard battery life;
- The main payload on a UAV is a directional antenna (*e.g.*, Yagi antenna) to capture radio signals;
- The signal processing module is a hardware component embodying a software defined radio capable of receiving and processing multiple radio-tag signals simultaneously.

The objects of interest are equipped with radio transmitters with on-off-keying signalling with low transmit power settings. This strategy is commonly used in numerous applications such as very high frequency (VHF) collared tags for wildlife tracking (Kays et al., 2011; Thomas, Holland and Minot, 2012; Cliff et al., 2015; Nguyen et al., 2019a), or safety beacons for search and rescue missions (Gerasenko et al., 2001; Murphy et al., 2008). The transmitter design and signalling methods are designed to conserve battery power, reduce the cost of the transmitters, increase the transmitters' lifespan as well as reduce installation and maintenance costs. Such a transmitter usually emits a pulse train of period $T_0$. Within this period, the pulse consists of a truncated sine wave with frequency $f$ over the interval $[\tau, \tau + P_w]$, as illustrated later in Figure 4.2. Low power on-off-keying signals are difficult to detect in noisy environments. The objects of interest, *e.g.*, people, wildlife, do not follow very predictable trajectories (such as cars, or planes), and most objects, wildlife, for instance, are afraid of the presence of the UAV in their territories. As a result, the UAV also needs to maintain a safe distance from objects, although getting close to the objects of interest improves tracking accuracy. Consequently, the received signals from the objects of interest are even harder to detect.

The problem we have articulated for involves tracking multiple radio-tagged objects of interest. The state of a single object of interest comprises of all of its kinematic state (denoted as $\zeta = [x, s]^T \in \mathbb{R}^4 \times \mathbb{S}$), including its position and velocity $x \in \mathbb{R}^4$, and its unknown dynamic model $s \in \mathbb{S}$ (*e.g.*, wandering, constant velocity). Furthermore, each object of interest transmits an on-off-keying signal, as illustrated later in Figure 4.2, with unknown offset time $\tau \in \mathbb{R}_0^+$ (a non-negative real number), and an unknown unique frequency index $\lambda \in \mathbb{L} \subset \mathbb{N}$ (a natural number). Thus, the state of a single object of interest is $\boldsymbol{x} = [\zeta, \tau, \lambda]^T \in \mathbb{T} = \mathbb{X} \times \mathbb{L}$, where $\mathbb{X} \subseteq \mathbb{R}^4 \times \mathbb{S} \times \mathbb{R}_0^+$.

### 4.3.2 Measurement model

Given a multi-object state $\boldsymbol{X} \in \mathcal{F}(\mathbb{T})$, each object $\boldsymbol{x} = [\zeta, \tau, \lambda]^T \in \boldsymbol{X}$, uniquely identified by frequency index $\lambda$, transmits an on-off-keying signal within a frequency band (*e.g.*, $148 - 152$ MHz VHF band commonly used for wildlife transmitters (Kenward, 2000)) to a directional antenna mounted on an observer.

The receiver model of the observer is illustrated in Figure 4.1. Here, a software defined radio (SDR) collects received signals from the antenna and down-converts

the received signal $v$ via the Hilbert transform and a mixer to a baseband signal $y$, which is subsequently digitised via an embedded analogue-to-digital converter (ADC) (Ossmann, 2015). The digitised signal is then transformed to the time-frequency domain via a short time Fourier transform (STFT) algorithm (Figure 4.1c). In practice, the following assumptions for the receiver are made:

- The required safety distance between the observer and each object of interest is sufficiently large, so that the transmitted signal can be treated as a far-field signal and the effect of multipath is negligible (Nguyen et al., 2019a);
- The receiver noise $\eta$, which may come from the outside environment or thermal noise generated from electronic devices within the receiver, is narrowband wide-sense-stationary (WSS) Gaussian because the bandwidth $B_w$ is small compared to the centre frequency $f_c$, $B_w \ll f_c$ (Orfanidis, 2002, pp.116).

In the following, we construct a model of the received signals captured by the receiver, beginning with the antenna model.

**Antenna Model** (Figure 4.1a): For a single object with state $\boldsymbol{x} = [\zeta, \tau, \lambda]^T$, the signal $s^{(\boldsymbol{x})}$ measured at a reference distance $d_0 > 0$ in the far field region can be modelled as:

$$s^{(\boldsymbol{x})}(t) = \frac{A^{(\lambda)}}{d_0^{\kappa}} \cos[2\pi(f_c + f^{(\lambda)})t + \phi^{(\lambda)}] \text{rect}_{P_w}^{T_0}(t - \tau), \tag{4.1}$$

where $A^{(\lambda)}, f^{(\lambda)}, \phi^{(\lambda)}$ are the signal amplitude, baseband frequency and phase, respectively, corresponding to frequency index $\lambda$ of object $\boldsymbol{x}$; $\kappa$ is a dimensionless path loss exponent that depends on the environment and typically ranges from 2 to 4; $f_c$ is the centre frequency of the band of interest; and

$$\text{rect}_{P_w}^{T_0}(t - \tau) = \sum_{n=-\infty}^{\infty} \text{boxcar}_{\tau}^{\tau + P_w}(t + nT_0) \tag{4.2}$$

is a periodic rectangular pulse train with period $T_0$ and pulse width $P_w$; $\text{boxcar}_a^b(\cdot)$ is a function which is unity on the interval $[a, b]$ and zero elsewhere.

At the output of the directional antenna, the noiseless received signal from a given set $\boldsymbol{X}$ of objects of interest is modelled as:

$$v^{(u)}(t) = \sum_{\boldsymbol{x} \in \boldsymbol{X}} v^{(\boldsymbol{x},u)}(t). \tag{4.3}$$

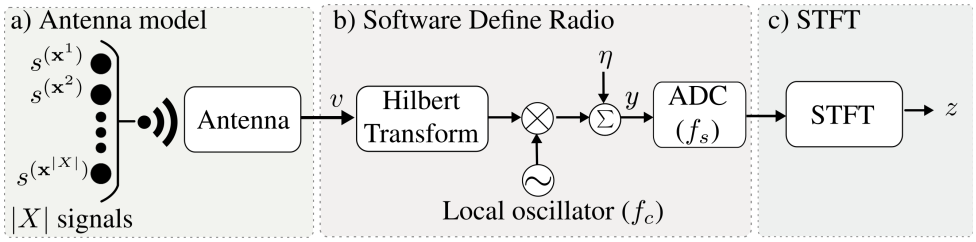


**Figure 4.1. The receiver model.** $|X|$ objects transmit on-off-keying analogue signals in the time domain. These signals are captured by the antenna and subsequently digitised through a software defined radio device, and converted to time-frequency domain measurements using an STFT algorithm.

Here, $v^{(\boldsymbol{x},u)}$ is the individual signal contribution of an object with state $\boldsymbol{x}$ measured by the observer with state $u$, given by (Nguyen et al., 2019a):

$$v^{(\boldsymbol{x},u)}(t) = \gamma(\zeta,u)\cos[2\pi(f_c + f^{(\lambda)})t + \psi(\zeta,u)]\mathrm{rect}^{T_0}_{P_w}(t-\tau), \tag{4.4}$$

where

- $u = [p^{(u)};\theta^{(u)}]$ is the observer state which comprises of its position $p^{(u)}$ and heading angle $\theta^{(u)}$;
- $\gamma(\zeta,u) = A^{(\lambda)}G_rG_a(\zeta,u)(d_0/d(p^{(\zeta)},p^{(u)}))^{\kappa}$ is the received signal magnitude when distance between the position of object $\boldsymbol{x}$ ($p^{(\zeta)}$) and the position of observer $u$ ($p^{(u)}$) is $d(p^{(\zeta)},p^{(u)})$;
- $G_r$ is the receiver gain to amplify the received signal;
- $G_a(\zeta,u)$ is the directional antenna gain that depends on a UAV's heading angle $\theta^{(u)}$ and its relative position with respect to the position of object $\boldsymbol{x}$;
- $\psi(\zeta,u) = \phi^{(\lambda)} - (f_c + f^{(\lambda)})d(p^{(\zeta)},p^{(u)})/c$ is the received signal phase, where $c$ is the signal velocity.

**Remark 1.** Notably, the measured signal $v^{(\boldsymbol{x},u)}$ always depends on the observer state $u$. Hereafter, for notational simplicity, $u$ is suppressed, *e.g.*, $v^{(\boldsymbol{x})} \triangleq v^{(\boldsymbol{x},u)}$, $\gamma(\zeta) \triangleq \gamma(\zeta,u)$.

**Software Defined Radio (SDR)** (Figure 4.1b): The received signal $v$ is down-converted from the VHF band to the baseband via the Hilbert transform and the mixer. This down-conversion step implemented on the SDR's hardware components is a linear operation and is presented here for completeness. The baseband signal, $\tilde{v}$, is given by:

$$\tilde{v}(t) = \sum_{\boldsymbol{x}\in\boldsymbol{X}} \tilde{v}^{(\boldsymbol{x})}(t), \tag{4.5}$$

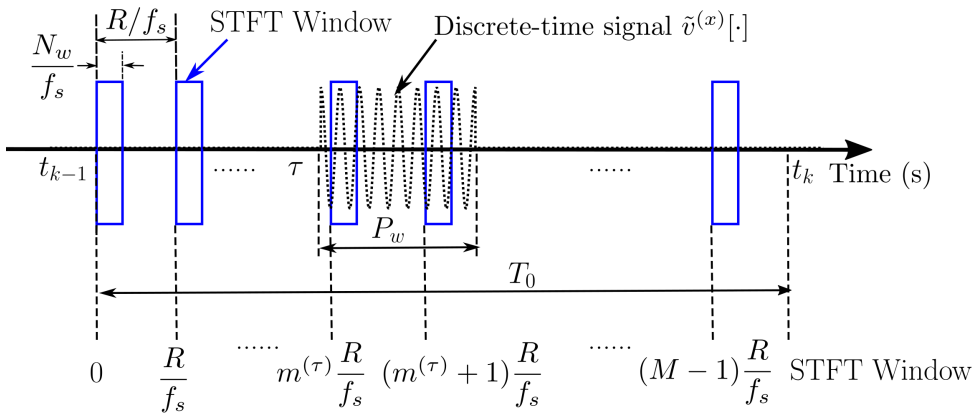


**Figure 4.2. Illustration for an on-off-keying discrete-time signal.** $\tilde{v}^{(x)}[\cdot]$ and a STFT windowing method at the $k^{\text{th}}$ measurement interval $[t_{k-1}, t_k)$. $R$ is the hop size, $N_w$ is the window width, $P_w$ is the pulse width, $\tau$ is the pulse time offset, $T_0$ is the period of the pulse. The STFT window frame is indexed at $mR/f_s$ where $m \in \{0, \ldots, M-1\}$, and $M$ is the number of window frames in one measurement interval. $m^{(\tau)} = \lceil \tau f_s / R \rceil$ is the time frame index of the signal transmitted from object $\boldsymbol{x}$.

where

$$\tilde{v}^{(\boldsymbol{x})}(t) \triangleq [v^{(\boldsymbol{x})}(t) + j[v^{(\boldsymbol{x})}]^*(t)]e^{-j2\pi f_c t} = \gamma(\zeta)e^{j\psi(\zeta)}e^{j2\pi f^{(\lambda)}t}\text{rect}_{P_w}^{T_0}(t-\tau), \tag{4.6}$$

$j$ is the imaginary unit; $[v^{(\boldsymbol{x})}]^*$ is the complex conjugate of $v^{(\boldsymbol{x})}$. Since the received signal is corrupted by receiver noise $\eta \sim \mathcal{N}(\cdot; 0, \Sigma_\eta)$, the total baseband signal $y$ can be written as:

$$y(t) = \sum_{\boldsymbol{x} \in \boldsymbol{X}} \tilde{v}^{(\boldsymbol{x})}(t) + \eta(t). \tag{4.7}$$

This continuous baseband signal $y(\cdot)$ in (4.7) is sampled at rate $f_s$ by the ADC component, which generates a discrete-time signal $y[\cdot]$, given by $y[n] \triangleq y(n/f_s)$.

**Short-Time Fourier Transform** (Figure 4.1c): The short time Fourier transform (STFT) converts the received signal to a time-frequency measurement. Since the on-off keying pulse offset time $\tau$ is unknown, we apply STFT to divide the measurement interval into shorter segments of equal length to capture the sinusoidal component of the received signal to estimate $\tau$ from the measurement. Figure 4.2 illustrates how the STFT is implemented over one measurement interval $[t_{k-1}, t_k)$ of a discrete on-off keying signal (the dash line in Figure 4.2) with period $T_0$ and pulse width $P_w$.

To capture the characteristics of the entire signal, we choose the $k^{\text{th}}$ measurement interval to be $[t_k - T_0, t_k)$ to fully contain one cycle of the periodic pulse train. The

discrete-time signal on $[t_k - T_0, t_k)$, at the STFT window frame $m \in \{0, \ldots, M-1\}$, is given by:

$$y_k^{(m)}[n] \triangleq y(t_k - T_0 + mR/f_s + n/f_s), \tag{4.8}$$

where $n = \{0, 1, \ldots, N_w - 1\}$.

We set the hop size $R$ and the STFT window width $N_w$ to meet the following condition,

$$1/f^{(\lambda)} \leq N_w < R = P_w f_s/2, \tag{4.9}$$

to ensure that the rectangular pulse of signal $\tilde{v}^{(x)}$ in (4.6) over the interval $[t_k - T_0 + \tau, t_k - T_0 + \tau + P_w)$ contains two non-overlapping STFT window indices, $\{m^{(\tau)}, m^{(\tau)} + 1\}$ as illustrated in Figure 4.2, such that these two STFT windows are only composed of the sinusoidal part of the signal. Thus, the number of window frames in one measurement interval is

$$M = \lceil 2T_0/P_w \rceil, \tag{4.10}$$

where $\lceil \cdot \rceil$ is the ceiling operator. The corresponding $L$-point STFT of $y_k^{(m)}[\cdot]$ using the windowing function $w[\cdot]$ is:

$$Y_k^{(m)}[l] = \sum_{n=0}^{N_w-1} y_k^{(m)}[n] w[n] e^{-j(n+mR)2\pi l/L}, \tag{4.11}$$

for $l = \{0, 1, \ldots, L-1\}$ (definitions of different window functions for extracting short-time signal segments and their properties can be found in (Smith III, 2011)).

At the $k^{\text{th}}$ measurement interval, let $\boldsymbol{X}_k$ denote the multi-object state and $\boldsymbol{x}_k = [\zeta_k, \tau_k, \lambda_k]^T$ be an element of $\boldsymbol{X}_k$. By substituting (4.2), (4.6), (4.7), (4.8) into (4.11), and combining with conditions in (4.9), $Y_k^{(m)}[l]$ can be written in term of signal and noise components as:

$$Y_k^{(m)}[l] = \sum_{\boldsymbol{x}_k \in \boldsymbol{X}_k} G^{(m,l)}(\boldsymbol{x}_k) + H_k^{(m)}[l], \tag{4.12}$$

where

$$G^{(m,l)}(\boldsymbol{x}_k) = \begin{cases} \gamma(\zeta_k) e^{j\psi(\zeta_k)} W[l - l^{(\lambda_k)}] & \text{if } m \in \{m^{(\tau_k)}, m^{(\tau_k)} + 1\}, \\ 0 & \text{otherwise,} \end{cases} \tag{4.13}$$

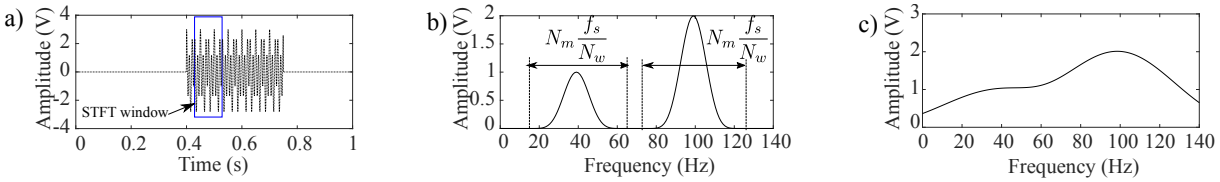


**Figure 4.3. Illustration for the resolvability of signal frequencies.** a) Two on-off-keying signals superpositioned in the time domain: $v(t) = [\cos(40t) + 2\cos(100t)]\text{rect}^1_{0.35}(t - 0.4)$ at the sampling rate $f_s = 1$ kHz; b) The signals are well-separated in frequency domain when using a 4-term Blackman Harris window with $N_m = 8$, where $N_w = 150$ samples and the main-lobe width (in Hz) $N_m f_s / N_w = 53.33$ Hz ¡ $\triangle f = 60$ Hz; c) However, it is not separable when $N_w = 42$ samples where the main-lobe width (in Hz) $N_m f_s / N_w = 190.47$ Hz ¿ $\triangle f = 60$ Hz.

$$W[m] = \sum_{n=0}^{N_w - 1} w[n] e^{-jn2\pi l/L}, \tag{4.14}$$

$$l^{(\lambda_k)} = \lfloor L f^{(\lambda_k)} / f_s \rfloor, \tag{4.15}$$

$$m^{(\tau_k)} = \lceil \tau_k f_s / R \rceil, \tag{4.16}$$

$$H_k^{(m)}[l] = \sum_{n=0}^{N_w - 1} \eta_k^{(m)}[n] w[n] e^{-j(n+mR)2\pi l/L}, \tag{4.17}$$

$$\eta_k^{(m)}[n] \triangleq \eta(t_k - T_0 + mR/f_s + n/f_s). \tag{4.18}$$

Now the measurement data $z_k$ at the $k^{\text{th}}$ measurement interval is an $M \times L$ matrix, with each element $z_k^{(m,l)} = |Y_k^{(m)}[l]|$, *i.e.*, the magnitude of $Y_k^{(m)}[l]$ defined in (4.12).

Notably, to increase the estimation accuracy of the number of transmitted signals, we need to reduce the interference among signal signatures in the frequency domain. Let $N_m$ denote the main-lobe width (in bins), where each windowing function $w[\cdot]$ affects $N_m$ differently, as shown in Table 4.1 (Smith III, 2011). Denote $\triangle f$ as the minimum frequency separation among all transmitted signals, given by $\triangle f = \min_{i,j \in \{1,\ldots,|\mathbf{X}|\}} |f^{(\lambda^i)} - f^{(\lambda^j)}|$ where $i \neq j$. To ensure resolvability of signal frequencies we require the main-lobe width (in Hz) of the signal signatures be well-separated (Smith III, 2011), as illustrated in Figure 4.3b; hence $N_m f_s / N_w \leq \triangle f$, which implies

$$N_w \geq \lceil N_m \frac{f_s}{\triangle f} \rceil. \tag{4.19}$$

Next, we derive the measurement likelihood given measurement $z_k$ and the condition in (4.19).

Table 4.1. Main-lobe width (in bins) $N_m$ for various windowing functions

| **Windowing Function** | Rectangular | Hamming | Blackman | $B$-term Blackman-Harris |
|---|---|---|---|---|
| $N_m$ | 2 | 4 | 6 | $2B$ |

### 4.3.3 Measurement likelihood function

Let $C(\boldsymbol{x}_k)$ denote the influence region of an object with state $\boldsymbol{x}_k$, given by:

$$C(\boldsymbol{x}_k) \triangleq \{(m,l) : |G^{(m,l)}(\boldsymbol{x}_k)| > 0\}, \tag{4.20}$$

where $G^{(m,l)}(\boldsymbol{x}_k)$ is defined in (4.13). We have the following proposition:

**Proposition 4.1.** *Given a multi-object $\boldsymbol{X}_k$, and its corresponding measurement $z_k$ at the $k^{th}$ measurement interval. If the influence region of each object does not overlap, i.e.,*

$$C(\boldsymbol{x}_k) \cap C(\boldsymbol{x}'_k) = \emptyset \; \forall \; \boldsymbol{x}_k, \boldsymbol{x}'_k \in \boldsymbol{X}_k, \tag{4.21}$$

*then the measurement likelihood function is given by:*

$$g(z_k|\boldsymbol{X}_k) \propto \prod_{\boldsymbol{x}_k \in \boldsymbol{X}_k} g_{z_k}(\boldsymbol{x}_k), \tag{4.22}$$

*where*

- $g_{z_k}(\boldsymbol{x}_k) = \prod_{(m,l)\in C(\boldsymbol{x}_k)} \dfrac{\varphi(z_k^{(m,l)}; |G^{(m,l)}(\boldsymbol{x}_k)|, \Sigma_z)}{\phi(z_k^{(m,l)}; \Sigma_z)}$,
- $\varphi(\cdot; |G^{(m,l)}(\boldsymbol{x}_k)|, \Sigma_z)$ *is the Ricean distribution with mean* $|G^{(m,l)}(\boldsymbol{x}_k)|$ *and covariance* $\Sigma_z$,
- $\phi(\cdot; \Sigma_z)$ *is the Rayleigh distributions with covariance* $\Sigma_z$,
- $\Sigma_z = E_w \Sigma_\eta / 2$ *is the receiver noise covariance in frequency domain,*
- $E_w = \sum_{n=0}^{N_w - 1} |w[n]|^2$ *is the window energy.*

The following Lemma facilitates the proof of Proposition 4.1.

**Lemma 4.2.** *The STFT of the discrete-time signal $y_k^{(m)}[\cdot]$ can be expressed in terms of in-phase and quadrature forms:*

$$Y_k^{(m)}[l] = \sum_{\boldsymbol{x}_k \in \boldsymbol{X}_k} G^{(m,l)}(\boldsymbol{x}_k) + H_k^{(m)}[l] = Y_{k,I}^{(m)}[l] + jY_{k,Q}^{(m)}[l]. \tag{4.23}$$

*Furthermore, the components, $Y_{k,I}^{(m)}[\cdot]$ and $Y_{k,Q}^{(m)}[\cdot]$, are independent non-zero mean Gaussian random variables with covariance $\Sigma_z = E_w \Sigma_\eta / 2$.*

**Proof:** First, we show that the in-phase and quadrature components of the noise terms $H_k^{(m)}[\cdot]$ of $\Upsilon_k^{(m)}[\cdot]$ are independent. Next, we prove that the magnitude of the signal term $\sum_{\boldsymbol{x}_k \in \boldsymbol{X}_k} G^{(m,\cdot)}(\boldsymbol{x}_k)$ of $\Upsilon_k^{(m)}[\cdot]$ has the form $\left|\mu W[\cdot]\right|$ where $\mu$ is zero or a constant and W is as defined in (4.14). Therefore, for a given frequency frame $l$, the in-phase and quadrature components of $\Upsilon_k^{(m)}[\cdot]$ are characterised by constant signal terms of the form $\left|\mu W[\cdot]\right|$ and independent noise terms. Thus, as proven in (Richards, 2013, pp.17), the in-phase and quadrature components are independent since their cross-correlation coefficient is zero. Detailed as below.

Since the receiver noise $\eta \sim \mathcal{N}(\cdot; 0, \Sigma_\eta)$ is narrowband wide-sense-stationary Gaussian, it can be rewritten in terms of in-phase and quadrature noise components (Davenport and Root, 1987, pp.159):

$$\eta(t) = \eta_I(t) + j\eta_Q(t). \tag{4.24}$$

where $\eta_I(\cdot)$ and $\eta_Q(\cdot)$ are independent zero mean Gaussian random variables with covariance $\Sigma_\eta/2$. Then the STFT transformation of the noise components into time-frequency frames in (4.17) follows:

$$H_k^{(m)}[l] = H_{k,I}^{(m)}[l] + jH_{k,Q}^{(m)}[l], \tag{4.25}$$

where $H_{k,I}^{(m)}[\cdot]$ and $H_{k,Q}^{(m)}[\cdot]$ are also independent zero-mean Gaussian random variables with covariance $\Sigma_z = E_w\Sigma_\eta/2$, as proven in (Richards, 2013, pp.10-12). Thus, by rewriting $\Upsilon_k^{(m)}[\cdot]$ in (4.12) in terms of in-phase and quadrature components, and letting $\Gamma_k^{(m)}[l] = \sum_{\boldsymbol{x}_k \in \boldsymbol{X}_k} G^{(m,l)}(\boldsymbol{x}_k)$, for simplicity, we have:

$$\begin{aligned}
\Upsilon_k^{(m)}[l] &= \Gamma_k^{(m)}[l] + H_k^{(m)}[l], && (4.26)\\
&= \left(\mathrm{Re}\{\Gamma_k^{(m)}[l]\} + H_{k,I}^{(m)}[l]\right) + j\left(\mathrm{Im}\{\Gamma_k^{(m)}[l]\} + H_{k,Q}^{(m)}[l]\right), && (4.27)\\
&= \Upsilon_{k,I}^{(m)}[l] + j\Upsilon_{k,Q}^{(m)}[l]. && (4.28)
\end{aligned}$$

From the initial assumption in Proposition 4.1, $C(\boldsymbol{x}_k) \cap C(\boldsymbol{x}'_k) = \emptyset \; \forall \; \boldsymbol{x}_k, \boldsymbol{x}'_k \in \boldsymbol{X}_k$. Thus, $(m,l) \notin C(\boldsymbol{x}_k) \cap C(\boldsymbol{x}'_k)$. In other words, at time-frequency frame $(m,l)$, at most one object $\boldsymbol{x}_k \in \boldsymbol{X}_k$ contributes to the magnitude of $|\Gamma_k^{(m)}[l]|$, such that:

$$|\Gamma_k^{(m)}[l]| = |\sum_{\boldsymbol{x}_k \in \boldsymbol{X}_k} G^{(m,l)}(\boldsymbol{x}_k)| = \begin{cases} |G^{(m,l)}(\boldsymbol{x}_k)| & \text{if } (m,l) \in C(\boldsymbol{x}_k), \\ 0 & \text{otherwise,} \end{cases} \tag{4.29}$$

where, following the signal model illustrated in Figure 4.2,

$$|G^{(m,l)}(\boldsymbol{x}_k)| = \begin{cases} \left|\gamma(\zeta_k)W[l-l^{(\lambda_k)}]\right| & \text{if } m \in \{m^{(\tau_k)}, m^{(\tau_k)}+1\} \\ 0 & \text{otherwise,} \end{cases} \tag{4.30}$$
$$C(\boldsymbol{x}_k) = \{m^{(\tau_k)}, m^{(\tau_k)}+1\} \times S(l^{(\lambda_k)}),$$

and $S(l^{(\lambda_k)}) \subseteq \{0,\ldots,L-1\}$ denotes the window function—see Table 4.1—dependent number of frequency samples contributed by object $\boldsymbol{x}_k$ .

According to (4.29) and (4.30), $|\Gamma_k^{(m)}[\cdot]|$ is deterministic and has the form $\left|\mu W[\cdot]\right|$, where $\mu$ is zero or a constant. Consequently, the cross-correlation coefficient $\rho_{IQ}$ of $Y_{k,I}^{(m)}[\cdot]$ and $Y_{k,Q}^{(m)}[\cdot]$ is zero, as proven in (Richards, 2013):

$$\rho_{IQ} = \left(\mathbb{E}(Y_{k,I}^{(m)}[l]Y_{k,Q}^{(m)}[l]) - \Gamma_{k,I}^{(m)}[l]\Gamma_{k,Q}^{(m)}[l]\right)/\Sigma_z = 0. \tag{4.31}$$

Therefore, $Y_{k,I}^{(m)}[\cdot]$ and $Y_{k,Q}^{(m)}[\cdot]$ are both independent non-zero mean Gaussian with the same covariance $\Sigma_z = E_w\Sigma_\eta/2$. □

**Proof of Proposition 4.1:** Applying Lemma 4.2, for any time-frequency frame $(m,l)$, $Y_{k,I}^{(m)}[l]$ and $Y_{k,Q}^{(m)}[l]$ are independent non-zero mean Gaussian. Thus, combining the result in (Richards, 2013, pp.17-18), if object $\boldsymbol{x}_k$ contributes to the measurement $z_k$ at time-frequency frame $(m,l)$: $|\Gamma_k^{(m)}[l]| = |G^{(m,l)}(\boldsymbol{x}_k)|$, then the measurement likelihood function of $z_k^{(m,l)} = |Y_k^{(m)}[l]|$ is:

$$p(z_k^{(m,l)}|\boldsymbol{x}_k) = \varphi(z_k^{(m,l)}; |G^{(m,l)}(\boldsymbol{x}_k)|, \Sigma_z), \tag{4.32}$$

where $\varphi(x;\nu,\Sigma) = x\exp\{-(x^2+\nu^2)/(2\Sigma)\}I_0(x\nu/\Sigma)/\Sigma$ is a *Ricean* distribution; $I_0(\cdot)$ is the Bessel function of the first kind defined as $I_0(x) = \sum_{j=0}^{\infty}(-1)^j(x^2/4)^j/(j!)^2$.

When no signal contributes to a frame $(m,l)$, $|\Gamma_k^{(m)}[l]| = 0$, then the measurement likelihood function of $z_k^{(m,l)}$ is:

$$p(z_k^{(m,l)}|\boldsymbol{x}_k) = \phi(z_k^{(m,l)}; \Sigma_z), \tag{4.33}$$

where $\phi(x;\Sigma) = x\exp\{-x^2/\Sigma\}/\Sigma$ is a *Rayleigh* distribution.

Thus, at any given frame $(m,l) \in \{0,\ldots,M-1\} \times \{0,\ldots,L-1\}$, the measurement likelihood function of $z_k^{(m,l)} = |Y_k^{(m)}[l]|$, given object state $\boldsymbol{x}_k$ follows:

$$p(z_k^{(m,l)}|\boldsymbol{x}_k) = \begin{cases} \varphi(z_k^{(m,l)}; |G^{(m,l)}(\boldsymbol{x}_k)|, \Sigma_z) & (m,l) \in C(\boldsymbol{x}_k), \\ \phi(z_k^{(m,l)}; \Sigma_z) & (m,l) \notin C(\boldsymbol{x}_k). \end{cases} \tag{4.34}$$

Since there is no overlap between the influence regions of two objects, *i.e.*, $C(\boldsymbol{x}_k) \cap C(\boldsymbol{x}'_k) = \emptyset \ \forall \ \boldsymbol{x}_k, \boldsymbol{x}'_k \in \boldsymbol{X}_k$, the measurement likelihood of $z_k$ conditioned on the multi-object state $\boldsymbol{X}_k$, can be modelled as a separable function:

$$g(z_k|\boldsymbol{X}_k) = \Big( \prod_{\boldsymbol{x}_k \in \boldsymbol{X}_k} \prod_{(m,l) \in C(\boldsymbol{x}_k)} \varphi(z_k^{(m,l)}; |G^{(m,l)}(\boldsymbol{x}_k)|, \Sigma_z) \Big) \prod_{(m,l) \notin \cup_{\boldsymbol{x}_k \in \boldsymbol{X}_k} C(\boldsymbol{x}_k)} \phi(z_k^{(m,l)}; \Sigma_z) \tag{4.35}$$

$$= \prod_{(m,l)=(0,0)}^{(M-1,L-1)} \phi(z_k^{(m,l)}; \Sigma_z) \prod_{\boldsymbol{x}_k \in \boldsymbol{X}_k} g_{z_k}(\boldsymbol{x}_k) \propto \prod_{\boldsymbol{x}_k \in \boldsymbol{X}_k} g_{z_k}(\boldsymbol{x}_k),$$

where

$$g_{z_k}(\boldsymbol{x}_k) = \prod_{(m,l) \in C(\boldsymbol{x}_k)} \frac{\varphi(z_k^{(m,l)}; |G^{(m,l)}(\boldsymbol{x}_k)|, \Sigma_z)}{\phi(z_k^{(m,l)}; \Sigma_z)}. \square$$

For our particular problem, given a multi-object $\boldsymbol{X}$, a single object $\boldsymbol{x} = [\zeta, \ \tau, \ \lambda]^T \in \boldsymbol{X}$ is uniquely identified by the unique frequency index $\lambda$. Furthermore, condition (4.19) ensures negligible interference in the frequency domain between the signals emitted from objects with different $\lambda$. As shown in (Harris, 1978), using the 4-term Blackman Harris window, the side-lobe level is less than $-92$ dB compared to the main-lobe level. Consequently, for all practical purposes, we can consider that the influence region of each object does not overlap, *i.e.*, $C(\boldsymbol{x}) \cap C(\boldsymbol{x}') = \emptyset \ \forall \ \boldsymbol{x}, \boldsymbol{x}' \in \boldsymbol{X}$. Thus, Proposition 4.1 applies to our measurement model.

### 4.3.4 Multi-object tracking

Tracking an unknown number of objects of interest under noisy measurements is a difficult problem. It is even more challenging when the number of objects of interest may change over time. Due to the low power characteristics of signals from radio-tagged objects, detection-based approaches often fail to detect objects in low signal-to-noise ratio (SNR) environments, especially when objects appear or disappear frequently, which lead to higher tracking errors. Thus, detection based approaches may not be suitable for tracking radio-tagged objects in low SNR environments due to the information loss during the thresholding process to detect objects' signals. On the other hand, the TBD method, using raw received signals as measurements, preserves all of the signals' information and has been successfully proven to be an effective filter under low SNR environments in (Barniv, 1985; Tonissen and Bar-Shalom, 1998; Rutten,

Gordon and Maskell, 2005; Buzzi, Lops and Venturino, 2005; Buzzi et al., 2008; Vo et al., 2010; Lehmann, 2012; Dunne and Kirubarajan, 2013; Papi et al., 2013, 2015).

We propose using the TBD-LMB filter (Papi et al., 2015) to track multiple, unknown and time-varying number of objects. For our particular problem, the single object state $\boldsymbol{x} = [\zeta, \tau, \lambda]^T = [\bar{\zeta}, \lambda]^T \in \boldsymbol{X}$ is uniquely identified by $\lambda \in \mathbb{L}$, where $\mathbb{L}$ (assumed to be known)[4] is a discrete label space containing all frequency indices $\lambda$, and $\bar{\zeta} = [\zeta, \tau]^T \in \mathbb{X}$ is the object state without label. Hence, the multi-object $\boldsymbol{X} \in \mathcal{F}(\mathbb{T})$ is in fact a labelled RFS. Our initial prior is an LMB density with label space $\mathbb{L}$ and an LMB birth model with label space $\mathbb{B}$ to accommodate an increase in the label space that can occur during UAV path planning for tracking objects[5]. Since we use the LMB birth model, TBD-GLMB filter in (Papi et al., 2015) reduces to a TBD-LMB filter.

TBD-LMB filter provides a simple and elegant solution for a multi-object tracking approach in a low SNR environment. However, existing applications of TBD-LMB filters do not make use of jump Markov system (JMS) models. Following (Reuter, Scheel and Dietmayer, 2015), we incorporate a JMS model to the proposed TBD-LMB filter by augmenting a discrete mode into the state vector: $\zeta = [x, s]^T$, where $x$ is the object position and velocity, $s \in \mathbb{S} = \{1, 2, ..., S_0\}$ is the object dynamic mode, $S_0 \in \mathbb{N}^+$ is a positive natural number. Moreover, the mode variable is modelled as first-order Markov chain with transitional probability $t_{k|k-1}(s_k|s_{k-1})$. Hence, the state dynamics and measurement likelihood for a single augmented state vector are given by:

$$\begin{aligned} \boldsymbol{\Phi}_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) &= \Phi_{k|k-1}(\bar{\zeta}_k|\bar{\zeta}_{k-1})\delta_{\lambda_{k-1}}(\lambda_k), \\ g_{z_k}(\boldsymbol{x}_k) &= g_{z_k}(x_k, \tau_k, \lambda_k) = g_{z_k}^{(\lambda_k)}(x_k, \tau_k), \end{aligned} \tag{4.36}$$

where

$$\Phi_{k|k-1}(\bar{\zeta}_k|\bar{\zeta}_{k-1}) = \mathcal{N}(x_k; F_{k-1}^{(s_{k-1})}x_{k-1}, Q^{(s_{k-1})})\mathcal{N}(\tau_k; \tau_{k-1}, Q^{(\tau)})t_{k|k-1}(s_k|s_{k-1}), \tag{4.37}$$

[4] In practice, the assumption that $\mathbb{L}$ is known holds; for example, conservation biologists possess a collection of radio-tagged wildlife captured, tagged and released back into the wild. However, $\lambda \in \mathbb{L}$ itself cannot be directly inferred from the measurements, especially under the low signal-to-noise ratio scenarios where existing object signals may or may not be received by the sensor and the sensor also receives interfering measurements (from other users) and thermal noise generated measurement artefacts not originating from any object.

[5] Notably, in an application where no new objects are introduced into the system over time, the label space $\mathbb{L}$ remains unchanged and the set of LMB birth parameters as expressed in (4.39) vanishes. In a practical application, the birth model can accommodate, for example, newly released wildlife during the operation of a tracking task by a UAV.

$\mathcal{N}(\cdot;\mu,Q)$ denotes a Gaussian density with mean $\mu$ and covariance $Q$, $F_{k-1}^{(s_{k-1})}$ is the single-object dynamic kernel on the discrete mode $s_{k-1}$. The offset time $\tau$ is estimated using a zero mean Gaussian random walk method with covariance $Q^{(\tau)} = \sigma_\tau^2 T_0^2$, where $\sigma_\tau^2$ is the standard deviation of the time offset noise. The frequency index $\lambda_k \in \mathbb{L}$ is unique and static, thus the transition kernel for $\lambda_k$ is given by:

$$\delta_{\lambda_{k-1}}(\lambda_k) = \begin{cases} 1 & \text{if } \lambda_k = \lambda_{k-1}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.38}$$

**LMB Prediction:** At time $k-1$, suppose the filtering density $\boldsymbol{\pi}_{k-1}$ is an LMB RFS described by the parameter set $\{r_{k-1}^{(\lambda)}, p_{k-1}^{(\lambda)}\}_{\lambda\in\mathbb{L}_{k-1}}$ with state space $\mathbb{X}$ and label space $\mathbb{L}_{k-1}$ (for notational convenience, we use $\boldsymbol{\pi}_{k-1} = \{r_{k-1}^{(\lambda)}, p_{k-1}^{(\lambda)}\}_{\lambda\in\mathbb{L}_{k-1}}$ to denote the density of an LMB RFS), and the birth model is also an LMB RFS $\boldsymbol{\pi}_{B,k} = \{r_{B,k}^{(\lambda)}, p_{B,k}^{(\lambda)}\}_{\lambda\in\mathbb{B}_k}$ with state space $\mathbb{X}$ and label space $\mathbb{B}_k$ (with $\mathbb{L}_{k-1}\cap\mathbb{B}_k = \emptyset$), then the predicted multi-object density is also an LMB RFS $\boldsymbol{\pi}_{k|k-1} = \{r_{k|k-1}^{(\lambda)}, p_{k|k-1}^{(\lambda)}\}_{\lambda\in\mathbb{L}_{k|k-1}}$ with state space $\mathbb{X}$ and label space $\mathbb{L}_{k|k-1} = \mathbb{L}_{k-1}\cup\mathbb{B}_k$, given by (Reuter et al., 2014):

$$\boldsymbol{\pi}_{k|k-1} = \{r_{E,k|k-1}^{(\lambda)}, p_{E,k|k-1}^{(\lambda)}\}_{\lambda\in\mathbb{L}_{k-1}} \cup \{r_{B,k}^{(\lambda)}, p_{B,k}^{(\lambda)}\}_{\lambda\in\mathbb{B}_k}, \tag{4.39}$$

where

$$r_{E,k|k-1}^{(\lambda)} = r_{k-1}^{(\lambda)} \cdot \langle p_{k-1}^{(\lambda)}, p_{S,k}^{(\lambda)}\rangle, \tag{4.40}$$

$$p_{E,k|k-1}^{(\lambda)}(\bar{\zeta}) = \frac{\langle \Phi_{k|k-1}(\bar{\zeta}|\cdot), p_{k-1}^{(\lambda)} p_{S,k}^{(\lambda)}\rangle}{\langle p_{k-1}^{(\lambda)}, p_{S,k}^{(\lambda)}\rangle}, \tag{4.41}$$

and $\langle\cdot\rangle$ is the inner product calculated on the previous state $\bar{\zeta}_{k-1}$, given by:

$$\langle \alpha, \beta\rangle = \sum_s \int \alpha(x,\tau|s)\beta(x,\tau|s)d(x,\tau). \tag{4.42}$$

**LMB Update:** Given the predicted LMB $\boldsymbol{\pi}_{k|k-1} = \{r_{k|k-1}^{(\lambda)}, p_{k|k-1}^{(\lambda)}\}_{\lambda\in\mathbb{L}_{k|k-1}}$ defined in (4.39), and a separable measurement likelihood function as in (4.22), then the filtering LMB is given by (Vo et al., 2010):

$$\boldsymbol{\pi}_k = \{r_k^{(\lambda)}, p_k^{(\lambda)}\}_{\lambda\in\mathbb{L}_{k|k-1}}, \tag{4.43}$$

where

$$r_k^{(\lambda)} = \frac{r_{k|k-1}^{(\lambda)}\langle p_{k|k-1}^{(\lambda)}, g_{z_k}^{(\lambda)}\rangle}{1 - r_{k|k-1}^{(\lambda)} + r_{k|k-1}^{(\lambda)}\langle p_{k|k-1}^{(\lambda)}, g_{z_k}^{(\lambda)}\rangle}, \tag{4.44}$$

$$p_k^{(\lambda)} = \frac{p_{k|k-1}^{(\lambda)} g_{z_k}^{(\lambda)}}{\langle p_{k|k-1}^{(\lambda)}, g_{z_k}^{(\lambda)}\rangle}. \tag{4.45}$$

### 4.3.5 Path planning under constraints

We formulate the online UAV path planning problem for joint detection and tracking as a partially observable Markov decision process (POMDP) which has been proven as an efficient and optimal technique for trajectory planning problems (Kaelbling, Littman and Cassandra, 1998; Castañón and Carin, 2008). In the POMDP framework, the purpose of path planning is to find the optimal policy (*e.g.* a sequence of actions) to maximise the total expected reward (Gostar et al., 2017). Hence, we first focus on evaluating the reward functions. Second, we incorporate a void constraint to maintain a safe distance between the UAV and objects of interest.

#### Reward Functions for Path Planning

Let $\mathcal{A}_k \in \mathbb{A}$ denote a set of possible control vectors $a_k$ at time $k$. A common approach is to calculate an optimal action that maximises the total expected reward over a look ahead horizon $H$ (Ristic and Vo, 2010; Hoang and Vo, 2014; Beard et al., 2017)—see Section 2.5.1:

$$a_k^* = \underset{a_k \in \mathcal{A}_k}{\arg\max}\, \mathbb{E}\Big[\sum_{j=1}^{H} \gamma^{j-1} \mathcal{R}_{k+j}(a_k)\Big]. \tag{4.46}$$

Since an analytical solution for the expectation of (4.46) is not available in general, two popular alternatives are to use Monte Carlo integration (Ristic and Vo, 2010; Beard et al., 2017) or the predicted ideal measurement set (PIMS) as in (Ristic, Vo and Clark, 2011; Hoang and Vo, 2014; Gostar, Hoseinnezhad and Bab-hadiashar, 2016). Using PIMS, the computationally lower cost approach, we only generate one ideal future measurement at each measurement interval (Hoang and Vo, 2014; Gostar, Hoseinnezhad and Bab-hadiashar, 2016). Hence, instead of (4.46), the optimal action is defined by:

$$a_k^* = \underset{a_k \in \mathcal{A}_k}{\arg\max} \sum_{j=1}^{H} \gamma^{j-1} \hat{\mathcal{R}}_{k+j}(a_k), \tag{4.47}$$

where

$$\hat{\mathcal{R}}_{k+j}(a_k) = D(\boldsymbol{\pi}_{k+j}(\cdot|z_{1:k}, \hat{z}_{k+1:k+j}(a_k), \boldsymbol{\pi}_{k+j|k}(\cdot|z_{1:k})). \tag{4.48}$$

In (4.48), the predicted density $\boldsymbol{\pi}_{k+j|k}(\cdot|z_{1:k})$ is calculated by propagating the filtering density $\boldsymbol{\pi}_k(\cdot|z_{1:k})$ in (4.43) using the prediction step[6] in (4.40), (4.41) repeatedly, from

[6]The prediction step generally includes birth, death and object motion. For improving computational time and tractability, we limit this to object motion only as in (Beard et al., 2017).

time $k$ to $k+j$. In contrast, the filtering density $\boldsymbol{\pi}_{k+j}(\cdot|z_{1:k}, \hat{z}_{k+1:k+j}(a_k))$ is computed recursively by propagating $\boldsymbol{\pi}_k(\cdot|z_{1:k})$ in (4.43) from $k$ to $k+j$ using both prediction in (4.40), (4.41) and update steps in (4.44), (4.45) with the ideal measurement $\hat{z}_{k+1:k+H}(a_k)$. The ideal measurement $\hat{z}_{k+1:k+j}(a_k)$ is computed by the following steps (Hoang and Vo, 2014):

*i)* Sampling from the filtering density $\boldsymbol{\pi}_k(\cdot|z_{1:k})$ in (4.43),

*ii)* Propagating it to $k+j$ using the prediction step in (4.40), (4.41),

*iii)* Calculating the number of objects $\hat{n}_{k+j|k}$ and the estimated multi-object state $\hat{\boldsymbol{X}}_{k+j|k} = \{\hat{\boldsymbol{x}}_{k+j|k}^{(i)}\}_{i=1}^{\hat{n}_{k+j|k}}$,

*iv)* Simulating the ideal measurement at $k+j$ based on the measurement model in (4.12) with the estimated state $\hat{\boldsymbol{X}}_{k+j|k}$.

The number of LMB components for the predicted density $\boldsymbol{\pi}_{k+j|k}(\cdot|z_{1:k})$ and the filtering density $\boldsymbol{\pi}_{k+j}(\cdot|z_{1:k}, \hat{z}_{k+1:k+j}(a_k))$ are the same because the measurement likelihood function is separable. For notational simplicity, $\boldsymbol{\pi}_1 \triangleq \boldsymbol{\pi}_{k+j|k}(\cdot|z_{1:k})$ and $\boldsymbol{\pi}_2 \triangleq \boldsymbol{\pi}_{k+j}(\cdot|z_{1:k}, \hat{z}_{k+1:k+j}(a_k))$ are two LMB densities on $\mathbb{X}$ with the same label space $\mathbb{L}$ (see Section 2.3.4 for a definition of an LMB density), given by:

$$\boldsymbol{\pi}_1 = \{r_1^{(\lambda)}, p_1^{(\lambda)}\}_{\lambda\in\mathbb{L}}, \ \boldsymbol{\pi}_2 = \{r_2^{(\lambda)}, p_2^{(\lambda)}\}_{\lambda\in\mathbb{L}}, \tag{4.49}$$

and rewriting $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ in terms of LMB densities:

$$\boldsymbol{\pi}_1(\boldsymbol{X}) = \delta_{|\boldsymbol{X}|}(|\boldsymbol{\mathcal{L}}(\boldsymbol{X})|) w_1(\mathcal{L}(\boldsymbol{X})) p_1^{\boldsymbol{X}}, \tag{4.50}$$

$$\boldsymbol{\pi}_2(\boldsymbol{X}) = \delta_{|\boldsymbol{X}|}(|\mathcal{L}(\boldsymbol{X})|) w_2(\mathcal{L}(\boldsymbol{X})) p_2^{\boldsymbol{X}}. \tag{4.51}$$

Hence, evaluating $\hat{\mathcal{R}}_{k+j}(a_{k+j})$ requires calculating the divergence between the two LMB densities $\boldsymbol{\pi}_2$ and $\boldsymbol{\pi}_1$. We consider two candidate divergence measures: *i)* Rényi divergence, and *ii)* Cauchy-Schwarz divergence described in Section 2.5.1. However, given the non-linearity of our measurement likelihood, both divergence measures have no closed form solution. Therefore, we approximate the divergence between two LMB densities using Monte Carlo sampling. In contrast to (Gostar, Hoseinnezhad and Bab-hadiashar, 2016) where Monte Carlo sampling was used to approximate the first moment, we approximate the full distribution.

1) **Rényi Divergence Approximation** From the definition in Section 2.5.1, we have:

$$\begin{aligned} D_{\text{Rényi}}(\boldsymbol{\pi}_2, \boldsymbol{\pi}_1) &= \frac{1}{\alpha - 1} \log \int \boldsymbol{\pi}_2^{\alpha}(\boldsymbol{X}) \boldsymbol{\pi}_1^{1-\alpha}(\boldsymbol{X}) \delta \boldsymbol{X} \\ &= \frac{1}{\alpha - 1} \log \int \Big[ \big(\delta_{|\boldsymbol{X}|}(|\boldsymbol{\mathcal{L}}(\boldsymbol{X})|) w_2(\mathcal{L}(\boldsymbol{X})) [p_2(\cdot)]^{\boldsymbol{X}}\big)^{\alpha} \\ &\qquad \times \big(\delta_{|\boldsymbol{X}|}(|\boldsymbol{\mathcal{L}}(\boldsymbol{X})|) w_1(\mathcal{L}(\boldsymbol{X})) [p_1(\cdot)]^{\boldsymbol{X}}\big)^{1-\alpha} \Big] \delta \boldsymbol{X}. \end{aligned} \tag{4.52}$$

Since $[p^{\boldsymbol{X}}]^{\alpha} = \left[\prod_{\boldsymbol{x} \in \boldsymbol{X}} p(\boldsymbol{x})\right]^{\alpha} = \prod_{\boldsymbol{x} \in \boldsymbol{X}} [p(\boldsymbol{x})]^{\alpha} = [p^{\alpha}]^{\boldsymbol{X}}$, using Lemma 3 in (Vo and Vo, 2013), this becomes:

$$D_{\text{Rényi}}(\boldsymbol{\pi}_2, \boldsymbol{\pi}_1) = \frac{1}{\alpha - 1} \log \left[ \sum_{L \subseteq \mathbb{L}} w_2^{\alpha}(L) w_1^{1-\alpha}(L) \prod_{\lambda \in L} \left[ \int [p_2^{(\lambda)}(\bar{\zeta})]^{\alpha} [p_1^{(\lambda)}(\bar{\zeta})]^{1-\alpha} d\bar{\zeta} \right] \right]. \tag{4.53}$$

Each $\lambda$ component of $\boldsymbol{\pi}_j$ $(j = 1, 2)$, the continuous density $p_j^{(\lambda)}(\cdot)$, is approximated by a probability mass function $\hat{p}_j^{(\lambda)}(\cdot)$ using the same set of samples $\{\bar{\zeta}^{(\lambda,i)}\}_{i=1}^{N_s}$ with different weights $\{\omega_j^{(\lambda,i)}\}_{i=1}^{N_s}$:

$$p_j^{(\lambda)}(\bar{\zeta}) \approx \hat{p}_j^{(\lambda)}(\bar{\zeta}) = \sum_{i=1}^{N_s} \omega_j^{(\lambda,i)} \delta_{\bar{\zeta}^{(\lambda,i)}}(\bar{\zeta}). \tag{4.54}$$

Using Monte Carlo sampling, the product between the two continuous densities in (4.53) can be approximated by the product of two probability mass functions on the finite samples $\{\bar{\zeta}^{(\lambda,i)}\}_{i=1}^{N_s}$, given by:

$$\begin{aligned} &\int [p_2^{(\lambda)}(\bar{\zeta})]^{\alpha} [p_1^{(\lambda)}(\bar{\zeta})]^{1-\alpha} d\bar{\zeta} \approx \sum_{i=1}^{N_s} [\hat{p}_2^{(\lambda)}(\bar{\zeta}^{(\lambda,i)})]^{\alpha} [\hat{p}_1^{(\lambda)}(\bar{\zeta}^{(\lambda,i)})]^{1-\alpha} \\ &\approx \sum_{i=1}^{N_s} \Big[ \sum_{j=1}^{N_s} \omega_2^{(\lambda,j)} \delta_{\bar{\zeta}^{(\lambda,j)}}(\bar{\zeta}^{(\lambda,i)}) \Big]^{\alpha} \Big[ \sum_{k=1}^{N_s} \omega_1^{(\lambda,k)} \delta_{\bar{\zeta}^{(\lambda,k)}}(\bar{\zeta}^{(\lambda,i)}) \Big]^{1-\alpha} \\ &\approx \sum_{i=1}^{N_s} [\omega_2^{(\lambda,i)}]^{\alpha} [\omega_1^{(\lambda,i)}]^{1-\alpha}. \end{aligned} \tag{4.55}$$

Substituting (4.55) into (4.53), the Rényi divergence becomes:

$$D_{\text{Rényi}}(\boldsymbol{\pi}_2, \boldsymbol{\pi}_1) \approx \frac{1}{\alpha - 1} \log \left[ \sum_{L \subseteq \mathbb{L}} w_2^{\alpha}(L) w_1^{1-\alpha}(L) \prod_{\lambda \in L} \left[ \sum_{i=1}^{N_s} (\omega_2^{(\lambda,i)})^{\alpha} (\omega_1^{(\lambda,i)})^{1-\alpha} \right] \right]. \tag{4.56}$$

2) **Cauchy-Schwarz Divergence Approximation** From the definition in Section 2.5.1 and following (Beard et al., 2017), we have:

$$D_{CS}(\boldsymbol{\pi}_2, \boldsymbol{\pi}_1) = -\log \left( \frac{\langle \boldsymbol{\pi}_2, \boldsymbol{\pi}_1 \rangle_K}{\sqrt{\langle \boldsymbol{\pi}_2, \boldsymbol{\pi}_2 \rangle_K \langle \boldsymbol{\pi}_1, \boldsymbol{\pi}_1 \rangle_K}} \right), \tag{4.57}$$

where

$$\langle \boldsymbol{\pi}_i, \boldsymbol{\pi}_j \rangle_K = \sum_{L \subseteq \mathbb{L}} w_i(L) w_j(L) \prod_{\lambda \in L} K\langle p_i^{(\lambda)}(\cdot), p_j^{(\lambda)}(\cdot) \rangle, \tag{4.58}$$

for $i, j \in \{1, 2\}$. Using the approach in (4.55), we have

$$\langle \boldsymbol{\pi}_i, \boldsymbol{\pi}_j \rangle_K \approx \sum_{L \subseteq \mathbb{L}} w_i(L) w_j(L) \prod_{\lambda \in L} K\Big( \sum_{k=1}^{N_s} \omega_i^{(\lambda,k)} \omega_j^{(\lambda,k)} \Big). \tag{4.59}$$

### Void Probability Functional

The UAV needs to maintain a safe distance from objects, although getting close to the objects of interest improves tracking accuracy. Therefore, in the following section, we derive a void constraint for the path planning formulation.

Let $V(u_{k+j}(a_k), r_{\min})$ denote the void region of objects based on a UAV's position at time $k+j$ if an action $a_k$ is taken. This leads to a cylinder shape where the ground distance between a UAV and any objects should be smaller than $r_{\min}$, given by:

$$V(u_{k+j}(a_k), r_{\min}) = \Big\{ \boldsymbol{x} \in \mathbb{X} : \sqrt{(p_x^{(\boldsymbol{x})} - p_x^{(u_{k+j}(a_k))})^2 + (p_y^{(\boldsymbol{x})} - p_y^{(u_{k+j}(a_k))})^2} < r_{\min} \Big\}, \tag{4.60}$$

where $p_x^{(\boldsymbol{x})}, p_y^{(\boldsymbol{x})}$ and $p_x^{(u_{k+j}(a_k))}, p_y^{(u_{k+j}(a_k))}$ denote positions of $\boldsymbol{x}$ and $u_{k+j}(a_k)$ in $x-y$ coordinates, respectively.

Using the closed form expression for the void probability functional[7] of the GLMB in (Beard et al., 2017), we impose the constraint in (44) on the trajectory planning problem as formulated below.

Given a region $S \subseteq \mathbb{X}$ and an LMB density $\boldsymbol{\pi}$ on $\mathbb{X}$ parameterised as $\boldsymbol{\pi} = \delta_{|\boldsymbol{X}|}(|\boldsymbol{\mathcal{L}}(\boldsymbol{X})|) w(\mathcal{L}(\boldsymbol{X})) p^{\boldsymbol{X}} = \{r^{(\lambda)}, p^{(\lambda)}\}_{\lambda \in \mathbb{L}}$ where each $\lambda$ component is approximated by a set of weighted samples $\{\omega^{(\lambda,i)}, \bar{\zeta}^{(\lambda,i)}\}_{i=1}^{N_s} : p^{(\lambda)}(\bar{\zeta}) \approx \sum_{i=1}^{N_s} \omega^{(\lambda,i)} \delta_{\bar{\zeta}^{(\lambda,j)}}(\bar{\zeta})$, the void functional of $S$ given the multi-object density $\boldsymbol{\pi}$, $B_\pi(S)$, can be approximated as:

$$B_{\boldsymbol{\pi}}(S) \approx \sum_{L \subseteq \mathbb{L}} w(\mathcal{L}) \prod_{\lambda \in \mathbb{L}} \Big(1 - \sum_{i=1}^{N_s} w^{(\lambda,i)} \delta_{\bar{\zeta}^{(\lambda,i)}}(\bar{\zeta}) 1_S(\bar{\zeta})\Big) \tag{4.61}$$

using the expression of the void probability functional in (Beard et al., 2017). Now the maximisation problem in (4.47) becomes:

$$a_k^* = \arg\max_{a_k \in \mathcal{A}_k} \sum_{j=1}^{H} \gamma^{j-1} \hat{\mathcal{R}}_{k+j}(a_k), \tag{4.62}$$

[7] Here, we use the notion of void probabilities as defined in (Kendall, Mecke and Stoyan, 1995).

subject to the constraint

$$\min_{j\in\{1,\ldots,H\}} [B_{\pi_{k+j}(\cdot|z_{1:k})}(V(u_{k+j}(a_k), r_{\min}))] > P_{\text{vmin}},$$

where $P_{\text{vmin}}$ denotes a void probability threshold.

### 4.3.6 Computational complexity analysis

In this subsection, we analyse the computational complexity of our proposed TBD-LMB tracking and planning algorithm. Since the planning algorithm consumes the most of computational time, we focus on analysing its computational complexity. In particular, the proposed planning algorithm consists of the following nested components:

i) Computation of raw measurements using STFT algorithms: $\mathcal{O}(MN_w \log(N_w))$,

ii) Computation of the information-based divergence: $\mathcal{O}(|\mathbb{L}|2^{|\mathbb{L}|}N_s)$,

iii) Computation of the optimal action $a_k^*$: $\mathcal{O}(|A|H)$.

Therefore, the total computational complexity of our proposed planning algorithm is $\mathcal{O}(MN_w \log(N_w)|\mathbb{L}|2^{|\mathbb{L}|}N_s|A|H)$. We can see that the computational complexity is proportional to $2^{|\mathbb{L}|}$, the hypothesis truncation method in (Vo, Vo and Phung, 2014) can be adopted to discard hypotheses with small weights to significantly improve the computational time. We can further improve the computational time by paralleling all of the above components using graphics processing units (GPUs).

## 4.4 Simulation Experiments

In this section, we evaluate the proposed online path planning strategy for joint detection and tracking of multiple radio-tagged objects using a UAV.

### 4.4.1 Experimental settings

A two-dimensional area of $[0, 1500]$ m $\times$ $[0, 1500]$ m is investigated to demonstrate the proposed approach. The UAV's height is maintained at 30 m while the objects' heights

are fixed at 1 m to limit the scope to a two-dimensional (2D) problem[8]. The total flight time is 400 s for all experiments.

We also follow the same practical constraints mentioned in (Nguyen et al., 2019a) for our simulations. The UAV cannot change its heading instantly, hence its maximum turning rate is limited to $\triangle\theta_k^u = |\theta_k^u - \theta_{k-1}^u| \leq \theta_{max}^u$ (rad/s). In addition, since the planning step normally consumes more time than the tracking step, we apply a cruder planning interval $N_p$ compared to measurement interval $T_0$, such that $N_p = nT_0$ where $n \geq 2$, $n \in \mathbb{N}$ (*i.e.*, $T_0 = 1$ s, $N_p = 5$ s, the planning algorithm calculates the best trajectory for the UAV in next five seconds at each five-measurement-intervals instead of every measurement-interval).

An object's dynamic mode $s$ follows the jump Markov system where its motion model is either: *i)* a *Wandering* (WD) mode where an object moves short distances without any clear purpose or direction, or *ii)* a constant velocity (CV) mode.

**The Wandering (WD) Model**:

$$x_k = F_{k-1}^{WD} x_{k-1} + q_{k-1}^{WD}, \tag{4.63}$$

where $F_{k-1}^{WD} = \text{diag}([1\ 0\ 1\ 0]^T)$, $q_{k-1}^{WD} \sim \mathcal{N}(0, Q^{WD})$ is a zero mean Gaussian process noise with covariance $Q^{WD} = \text{diag}([0.25\ \text{m}^2, 2.25\ (\text{m/s})^2,\ 0.25\ \text{m}^2, 2.25\ (\text{m/s})^2]^T)$. In the wandering model, the velocity components are instantly forgotten and then sampled from covariance $Q^{WD}$ at each time step $k$. However, the sampled velocity components do not influence an object's position. Further, the velocity components in $Q^{WD}$ are significantly larger than the position components therein. This is necessary to achieve the fast moving behaviour of objects in the constant velocity dynamic mode when an object switches from the wandering mode to the constant velocity mode.

**The Constant Velocity (CV) Model**:

$$\begin{aligned} x_k &= F_{k-1}^{CV} x_{k-1} + q_{k-1}^{CV}, \\ F_{k-1}^{CV} &= \begin{pmatrix} 1 & T_0 \\ 0 & 1 \end{pmatrix} \otimes I_2, \end{aligned} \tag{4.64}$$

[8]It can be easily extended to 3D; however, to save computational power, we limit our problem to the 2D domain.

**Table 4.2.** Birth, death, and dynamic mode parameters

| Parameter | Value |
|---|---|
| Birth probability ($r_B$) | $10^{-6}$ |
| Survival probability ($p_S$) | 0.99 |
| Initial mode probability | $[0.5\ 0.5]^T$ |
| Mode transitional probability | $[0.99\ 0.01; 0.01\ 0.99]$ |
| Constant velocity noise ($\sigma_{CV}$) | 0.05 m/s$^2$ |

where $\otimes$ denotes the Kronecker tensor product operator between two matrices, and $q_{k-1}^{CV} \sim \mathcal{N}(0, Q^{CV})$ is a $4 \times 1$ zero mean Gaussian process noise, with covariance

$$Q^{CV} = \sigma_{CV}^2 \begin{pmatrix} T_0^3/3 & T_0^2/2 \\ T_0^2/2 & T_0 \end{pmatrix} \otimes I_2,$$

and $\sigma_{CV}$ is the standard deviation of the process noise parameter.

There are four objects with different birth and death times, listed in pairs as $(t_{\text{birth}}, t_{\text{death}})$: $(1, 250)$, $(50, 300)$, $(100, 350)$, $(150, 400)$ s. The four objects initially follow the wandering model (WD) with initial state vectors $[800, 0.13, 300, -1.44]^T$, $[200, 0.18, 700, -2.17]^T$, $[1200, -1.94, 1000, 0.42]^T$, $[900, 1.91, 1300, -2.04]^T$ (with appropriate standard units) at birth. One second period after birth, object 1 and object 3 switch their dynamic mode to the constant velocity mode while object 2 and object 4 continue to follow the wandering model for 65 s. We detail the mode changes (later) in Figure 4.6.

For each newly born object, we assume an initial birth state described by a Gaussian distribution with means $[800, 0, 300, 0]^T$, $[200, 0, 700, 0]^T$, $[1200, 0, 1000, 0]^T$, $[900, 0, 1300, 0]^T$ (with appropriate standard units) and covariance $Q^B = \text{diag}([100\ \text{m}^2, 4\ (\text{m/s})^2, 100\ \text{m}^2, 4\ (\text{m/s})^2]^T)$. In practice, such a setting is reasonable and captures the prior knowledge about an object's location. For example, in applications such as wildlife tracking, conservation biologists know the location of newly released wildlife or the locations of entry and exit points of animals that can suddenly appear in a scene from underground animal dwellings.

**Table 4.3.** Signal parameters

| Parameter | Symbol | Value |
|---|---|---|
| Centre frequency | $f_c$ | 150 MHz |
| Baseband frequencies | $f^{(\lambda)}$ | 131 kHz, 201 kHz, 401kHz , 841 kHz |
| Sampling frequency | $f_s$ | 2 MHz |
| Pulse period | $T_0$ | 1 s |
| Pulse offset time | $\tau^{(\lambda)}$ | 0.1 s, 0.2 s, 0.3 s, 0.4 s |
| Pulse width | $P_w$ | 18 ms |
| Reference distance | $d_0$ | 1 m |
| Pulse amplitude | $A$ | 0.0059 V |
| Path loss constant | $\kappa$ | 3.1068 |

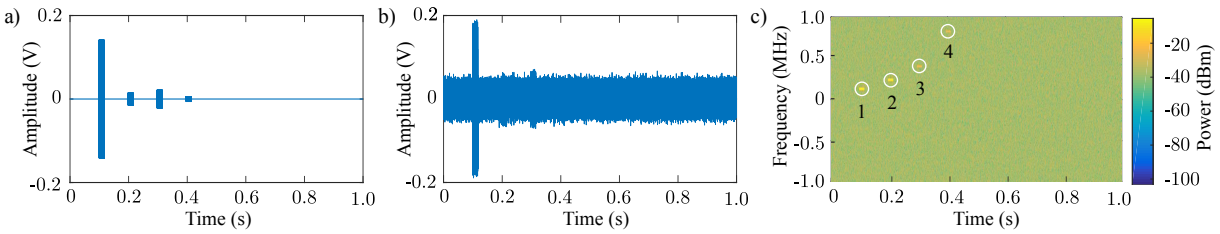


**Figure 4.4. An illustration of received signals.** For transmitting objects at distances of $[120,\ 515,\ 400,\ 920]$ m for object 1, object 2, object 3 and object 4 respectively, to the UAV in the presence of complex receiver noise covariance $\Sigma_\eta = 0.02^2\ V^2$. a) The received signal in the time domain without noise; b) The received signal in the time domain in the presence of the complex white noise; c) Spectrogram of the received signal in discrete time and frequency domain ($111 \times 256$ frames) where the bright spots represent an object's signal in a time-frequency frame.

The common parameters used in the following experiments are listed in Tables 4.2, 4.3, and 4.4. In addition, Figure 4.4a illustrates a raw received signal without noise from four transmitted objects along with a noisy received signal in Figure 4.4b. Furthermore, a single measurement set of the noisy received signal after going through the STFT process consisting of $111 \times 256$ time-frequency frames is illustrated in Figure 4.4c.

[9]The current void distance of 50 m is supported by our observations in (unpublished) field experiments involving UAV flights towards Southern Hairy-nosed wombats at Koolola station, South Australia to assess disturbances from UAVs.

**Table 4.4.** Measurement parameters

| Parameter | Symbol | Value |
|---|---|---|
| Receiver gain | $Gr$ | 72 dB |
| Receiver noise covariance | $\Sigma_\eta$ | $0.025^2$ V$^2$ |
| Number of window frames | $M$ | 111 |
| Number of frequency samples | $L$ | 256 |
| Window width | $N_w$ | 256 |
| Number of particles | $N_s$ | 50,000 |
| UAV's max heading angle | $\theta^u_{\max}$ | $\pi/3$ rad/s |
| UAV's velocity | $v_u$ | 20 m/s |
| UAV's initial position | $u_1$ | $[0, 0, 30, \pi/4]^T$ |
| Planning interval | $N_p$ | 5 s |
| Look-a-head horizon | $H$ | 3 |
| Minimum distance[9] | $r_{\min}$ | 50 m |
| Void threshold | $P_{\text{vmin}}$ | 0.9 |
| OSPA (order, cut-off) | $(p, c)$ | (1, 100 m) |

### 4.4.2 Experiments and results

We conduct two experiments: *i)* to validate and evaluate our proposed planning method for joint detection and tracking; *ii)* to compare performance against planning for tracking with conventional detection-then-track methods.

**Experiment 1–Validating Planning for Joint Detection and Tracking:** The first experiment is conducted with four objects in various locations and moving in different directions where birth and death times and motion dynamics are described in Section 4.4.1. We employ a Rényi divergence based reward function with receiver noise covariance $\Sigma_\eta = 0.025^2$ V$^2$ and the UAV undergoes trajectory changes every 15 s, *i.e.*, the planning interval $N_p = 5$ s with a look ahead horizon $H = 3$ (see Table 4.4). Figure 4.5a-b depict true object trajectories, birth and death times together with the estimated tracking accuracy for a typical experiment run. The results show that the

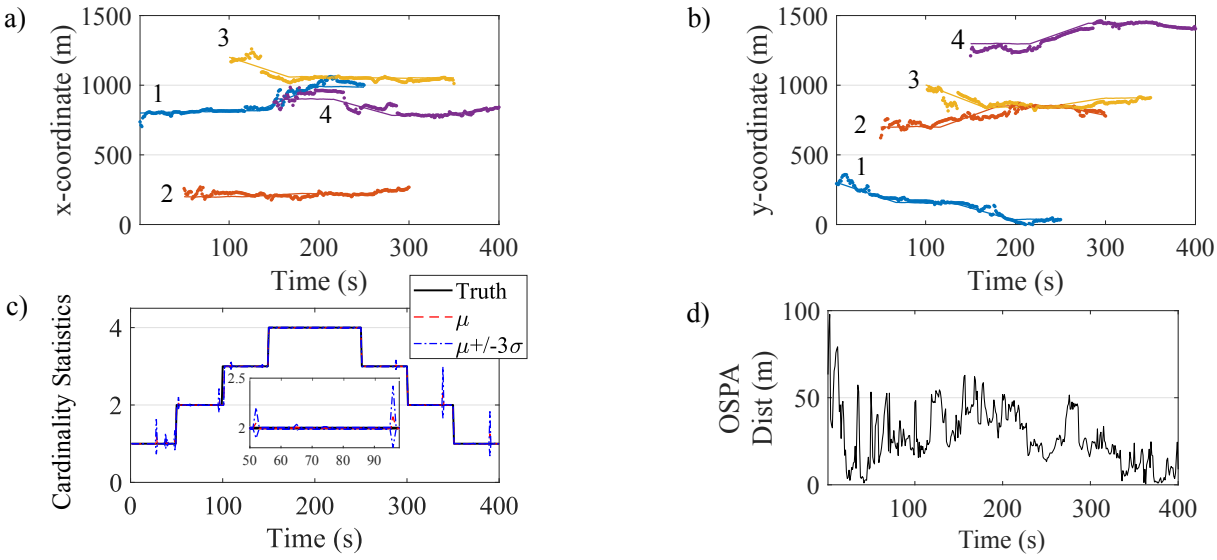


**Figure 4.5. Tracking four objects in various locations with different birth and death times and motion dynamics.** Estimated positions and truth in: a) x-coordinate; b) y-coordinate; c) cardinality—its truth versus mean $\mu$ and its variance $(\mu \pm 3\sigma)$; d) OSPA—the cut-off and order parameters are given in Table 4.4.

proposed planning for joint detection and tracking accurately estimates position and cardinality of the objects.

Figure 4.5c depicts the ground truth changes in the number of objects over time with the estimated cardinality. We used the optimal sub-pattern assignment (OSPA) metric (Schuhmacher, Vo and Vo, 2008) to quantify the error between the filter estimates and the ground truth to evaluate the multi-object miss distance. The spikes in Figure 4.5c indicate a high uncertainty in the estimated cardinality distribution. The high uncertainty is due to low signal-to-noise ratio (SNR) of received measurements. During path planning, noisy signals lead to poor control decisions that result in the UAV navigating to positions further from objects of interest where the signal incident on the UAV sensor antenna is often at an angle where the antenna gain is poor. Further, planning decisions are also subject to void constraints. Consequently, the existence probability of objects of interest can suddenly increase or decrease after a poor control action.

The OSPA error over the tracking period for these objects is depicted in Figure 4.5d. We see changes in the OSPA distance during birth and death events and their subsequent reduction as the planning algorithm undergoes course changes to improve tracking accuracy. These results confirm that our trajectory planning algorithm consistently

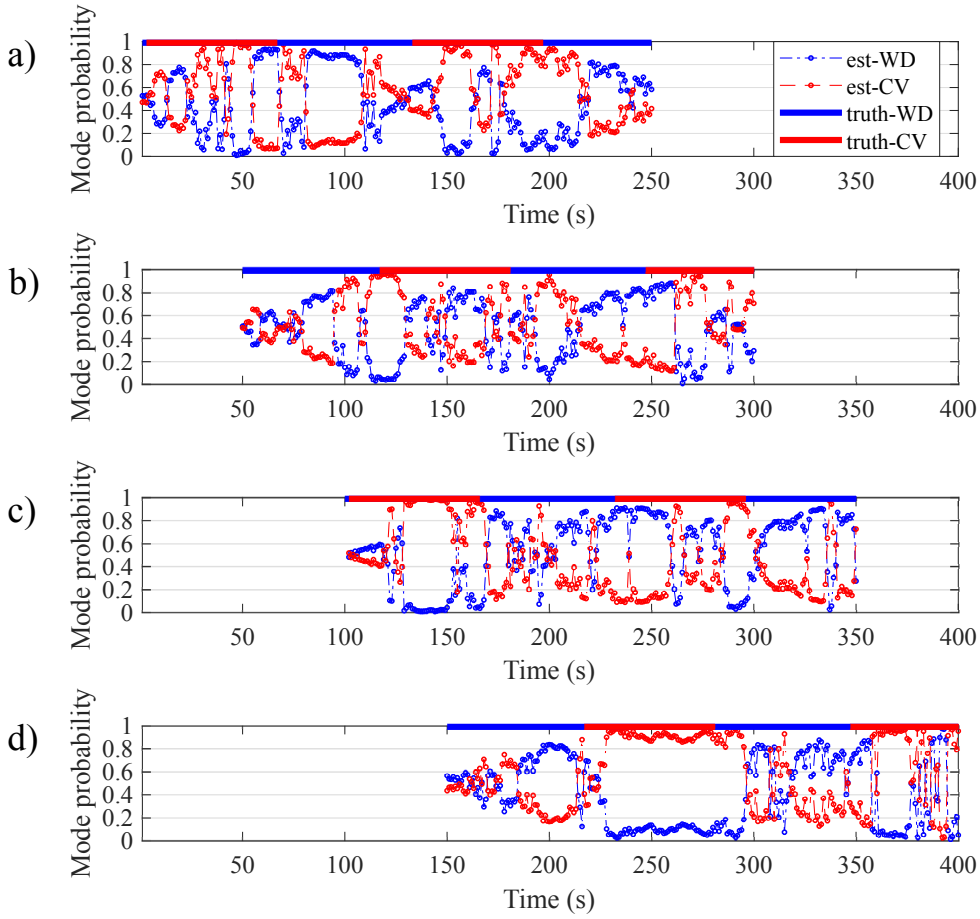


**Figure 4.6. The estimated mode probability for four objects.** Mode WD: Wandering and mode CV: Constant Velocity of a) object 1; b) object 2; c) object 3; and d) object 4.

tracks the time-varying number of objects over time whilst making course changes to improve estimation accuracy of all the objects.

Figure 4.6 depicts the multiple motion modes of objects and how it changes over time. The results show that although the received signals are noisy, the filter can still accurately estimate the correct mode of objects most of the time.

Figure 4.7 depicts the evolution of true and estimated object trajectories under the control of the path planning scheme subject to the void constraint. From these snapshots in time, we can see that a typical trajectory to track objects under the birth and death process agrees with our intuition. Initially, the UAV navigates towards object 1. At time $t = 50$ s (Figure 4.7a), object 2 is born; subsequently, the UAV maintains a trajectory between the two objects with course changes to track both objects. Object 3 is born at $t = 100$ s, and the UAV undertakes course changes to estimate the positions of all three moving objects with a manoeuvre to follow object 1

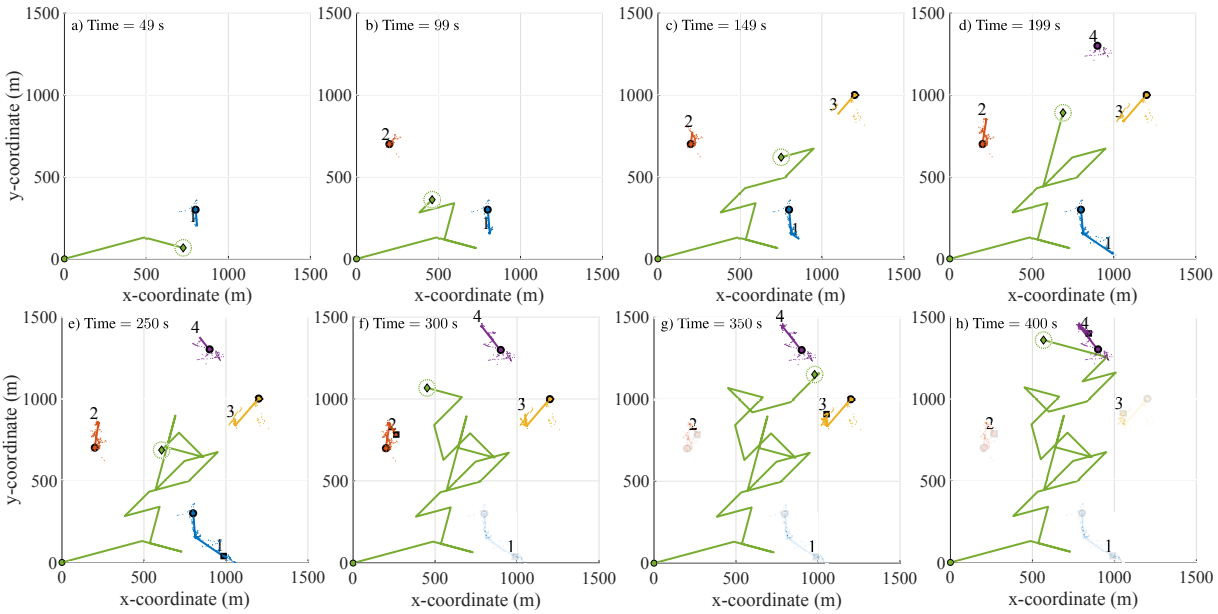


**Figure 4.7. A typical UAV trajectory.** A UAV trajectory (green path) under the proposed path planning for joint detection and tracking algorithm for multiple radio-tagged objects. Here: '○' denotes locations of object births; '□' denotes locations of object deaths; '◊' denotes current locations of the UAV. Faint tracks show objects subject to a death process.

and 2 whilst moving closer to object 3 (Figure 4.7bc). We can observe a similar planning strategy evolving when object 4 is born at time 150 s. The UAV navigates to a position to be closer to all four objects and maintain a position at the centre of the four objects to estimate the position of all four objects (Figure 4.7de). At time 250 s, object 1 vanishes, thus the UAV moves up towards a position at the centre of object 2, object 3 and object 4 to track the remaining objects (Figure 4.7f). Beyond 300 s, both object 1 and object 2 are no longer in existence; therefore we can observe the UAV heading to a position between objects 3—whilst maintaining the void constraint illustrated by the dashed circle at the UAV position—and object 4 (Figure 4.7g). After time 350 s, only object 4 exists; thus, the UAV undertakes trajectory changes to move towards object 4 (Figure 4.7h). The results show that the proposed planning strategy is able to detect and track all objects whilst dynamically acting upon different birth and death events to manoeuvre the UAV to move to positions that minimise the overall tracking error.

**Experiment 2–Comparing Performance:** In this experiment, we compare our proposed online path planning for joint detection and tracking formulation with the TBD-LMB filter with planning for detection-then-track (DTT) methods using a

DTT-LMB filter (Reuter et al., 2014). We compare three trajectory planning approaches for tracking: *i)* a straight path—direct the UAV back and forth along a diagonal line between $(0, 0)$ m and $(1500, 1500)$ m, *ii)* planning with Rényi divergence as the reward function, and *iii)* planning with Cauchy divergence as the reward function.

The measurements for DTT are extracted based on a peak detection algorithm to find the prominent peak such that the minimum peak separation is $N_m = 8$ frequency bins—i.e. the number of main-lobe width (in bins) for a 4-term Blackman Harris window as listed in Table 4.1. Since we examine the filter performance under various receiver noise levels, it is more appropriate to use a peak detection method compared to a fixed threshold value. Further, the peak detection method is robust against different noise levels, considering false-alarms and misdetections rates (Scholkmann, Boss and Wolf, 2012). The planning for DTT methods uses the same PIMS approach as per the TBD planning described in Section 4.3.5.

We use the OSPA distance and its cardinality component to compare performance across the three planning strategies for TBD and DTT approaches. We perform 100 Monte Carlo runs for each of the six cases, and receiver noise levels $\Sigma_\eta = 0.010^2, 0.015^2, \ldots, 0.050^2$ V$^2$ for the scenario shown in Figure 4.7. OSPA distance and cardinality results in Figure 4.8 show that the proposed path planning for TBD strategy provides significantly better estimation performance over planning for DTT-based strategies as demonstrated by the lower OSPA distance in the presence of increasing receiver noise. The TBD approaches are more effective than DTT approaches, especially due to the failure of DTT methods to detect changes in the number of objects in the presence of birth and death processes as evident in Figure 4.8b.

Intuition suggests that an information based approach should execute control actions to continually position the UAV to locations with the best ability to track multiple objects undergoing motion changes. Information based planning strategies outperforming the straight path approaches in both the TBD and DTT methods agrees with this intuition. Although, Rényi or Cauchy divergence as reward functions improve the overall tracking performance compared to the straight path method, we also observe that Rényi divergence is more discriminative than Cauchy divergence in our task and yields better OSPA distance values and hence the best performance.

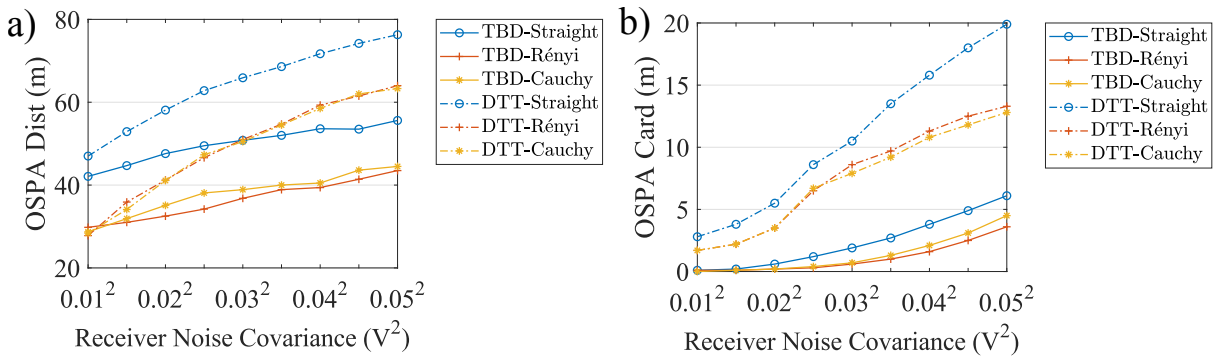


**Figure 4.8. A performance comparison between TBD and DTT.** Mean OSPA performance comparison across increasing receiver noise values. Here, -Straight, -Rényi and -Cauchy denote straight path, Rényi divergence and Cauchy divergence based planning strategies, respectively: a) OSPA distance; b) OSPA cardinality.

## 4.5 Conclusion

In this chapter, we have proposed an online path planning algorithm for joint detection and tracking of multiple radio-tagged objects under low SNR conditions. The planning for multi-object tracking problem was formulated as a POMDP with two information-based reward functions and the JMS TBD-LMB filter. In particular, the planning formulation incorporates a practical constraint to maintain a safe distance between the UAV and objects of interest to minimise the disturbances from the UAV. We have derived a measurement likelihood for the TBD-LMB filter and proved that the likelihood is separable in practice for multiple radio-tagged objects, thereby deriving an accurate multi-object TBD filter. The results demonstrated that our approach is highly effective in reducing the estimation error of multiple-objects in the presence of low signal-to-noise ratios compared to both detection-then-track approaches and tracking without planning.

Up until this chapter, we only considered the problem of tracking and planning using a single UAV to track multiple objects using a sensor with a large detection range. In reality, the on-board sensors mounted on the UAV are usually range-limited. Thus, to enlarge the search area and improve tracking accuracy, we must rely on multiple UAVs. In the next chapter, we will investigate the problem of controlling multiple UAVs to search for and track an unknown and time-varying number of objects with field of view (FoV) limited onboard sensors.

# Chapter 5

# Multi-Objective Multi-Agent Planning for Discovering and Tracking

WE consider the challenging problem of online planning for a team of agents to autonomously search and track a time-varying number of mobile objects under the practical constraint of detection range limited, or limited FoV, onboard sensors. A standard POMDP with a value function that either encourages discovery or accurate tracking of mobile objects is inadequate to *simultaneously* meet the *conflicting goals* of searching for undiscovered mobile objects whilst keeping track of discovered objects. Further, the planning problem is complicated by misdetections or false-detection of objects caused by range limited sensors and noise inherent to sensor measurements. We formulate a *novel* multi-objective POMDP based on information theoretic criteria, and an online multi-object tracking filter for the problem. Since controlling multi-agents is a well known combinatorial optimisation problem, assigning control actions to agents necessitates a greedy algorithm. We prove that our proposed multi-objective value function is a monotone submodular set function; consequently, the greedy algorithm can achieve a $(1 - 1/e)$ approximation for maximising the submodular multi-objective function.

## 5.1 Motivation and Contribution

We study the problem of controlling a team of agents to jointly track discovered mobile objects and explore the environment to search for undiscovered mobile objects of interest. Such problems are ubiquitous in wildlife tracking (Kays et al., 2011; Thomas, Holland and Minot, 2012; Cliff et al., 2015; Nguyen et al., 2019a), search and rescue missions (Gerasenko et al., 2001; Murphy et al., 2008). For instance, a team of unmanned aerial vehicles (UAVs) can be deployed to monitor activities of endangered radio-tagged wildlife in a survey scene, or to search for victims in a disaster response (Beck et al., 2018). Hence, it is critical to not only search for undiscovered objects but also track the movements of discovered objects of interest. Consequently, the overall team's objectives arise as a natural multi-objective optimisation problem, where several pertinent goals (*i.e.*, tracking and discovering) need to be simultaneously achieved.

Intrinsically, searching for undiscovered objects whilst simultaneously tracking visible objects are *competing* goals because, in practice, agent sensor systems, such as cameras, have a limited detection range. A single agent may only observe a small region of space and a decision to leave a visible object to explore hitherto unseen regions will lead to losing track of visible objects. Therefore, an agent observing a small region of the search area needs to collaboratively interact with other agents to *plan* its course of actions to collectively maximise the overall team's objectives of *tracking* and *discovering* multiple objects.

Multi-agent planning to achieve *multiple competing objectives* remains a challenging problem because of the complex interactions between agents leading to combinatorial optimisation problems (Wai et al., 2018). In practice, the problem is *further* complicated because: *i)* the agent sensors are not only limited in range but also sensitivity, and measurements are always subjected to environmental noise. Consequently, object detectors suffer from both missing detection of objects and false-detection; and *ii)* the number of objects of interest is often unknown, and varies with time since mobile objects can enter and leave the scene anytime (Vo et al., 2012). Most critically, the computation of optimal planning actions must be timely for real-world applications.

We propose a framework for multiple agents to jointly plan, search and track a time-varying number of objects using a novel multi-objective information-based value function formulation. Our multi-objective value function captures the competing objectives of planning for *tracking* and *discovery*. We adopt the random finite set (RFS)

model for the collection of objects of interest to account for the random appearance and disappearance of objects and their dynamics. Our proposed multi-objective value function maximises information gain over a look-ahead horizon for both discovered and undiscovered objects. Most importantly, our multi-objective value function is proven to be a monotone submodular set function; thus, we can cope with the intractability of the multi-objective optimisation problem (MOP) by employing a greedy algorithm. Our ability to use a greedy algorithm facilitates the computation of approximately optimal control actions with linear complexity in the number of agents for realising an online planning method.

## 5.2 Related Work

Multi-agent path planning in partially observable environments is a difficult problem for which the partially observable Markov decision processes (POMDP) approach has recently gained significant interest (Silver and Veness, 2010; Messias, Spaan and Lima, 2011; MacDermed and Isbell, 2013). Although the cooperation problem can be formulated as a decentralised POMDP (Dec-POMDP), its exact solutions are NEXP-complete (Bernstein et al., 2002). This is especially problematic for multi-agent POMDPs since the action and observation space grows exponentially with the number of agents (Amato and Oliehoek, 2015). One approach is distributed POMDPS (*e.g.*, networked distributed POMDP (Nair et al., 2005)) by exploiting interactions among neighbouring agents using distributed constraint optimisation. However, achieving a global goal for multi-agents in a distributed manner is an NEXP-problem in worst case scenarios (Rizk, Awad and Tunstel, 2018). To cope with this intractability, we adopt the MPOMDP centralised approach (Messias, Spaan and Lima, 2011) for controlling multiple agents (Dames and Kumar, 2015; Dames, Tokekar and Kumar, 2017; Wang et al., 2018).

POMDP has also been employed for sensor selection problems, *e.g.*, (Spaan, Veiga and Lima, 2015; Satsangi et al., 2018) proposed using the $\rho$POMDP (Araya et al., 2010) for a mobile agent to select $K$ in $N$ available sensors to search and track multiple objects. In particular, (Spaan, Veiga and Lima, 2015) proposed a method that always assumes the existence of one extra object in the scene to encourage discovery. However, biasing the cardinality estimate generates sub optimal planing decisions at the cost of tracking performance.

Our study focuses on the problem of controlling a team of agents for the task of tracking and discovering mobile targets. The task requires a suitable tracking framework. Studies on tracking objects have employed approaches such as multiple hypotheses tracking (MHT) (Reid, 1979) or joint probabilistic data association (JPDA) (Blackman and Popoli, 1999). The complex nature of our problem requires a framework that has the notion of probability of a random collection due to a time-varying and random number of objects where the states of objects are random vectors. The random finite set (RFS) (Mahler, 2007b) is the only framework that has the notion of probability density of a random set. Hence, we adopt RFS as our tracking framework.

Information-based path planning under the RFS framework for a single agent has been studied in several works (Ristic and Vo, 2010; Hoang and Vo, 2014; Beard et al., 2017). Most studies on multi-agent path planning using an RFS framework, are based on the generalised covariance intersection (GCI) methods with the assumption that agents have a consensus view of all objects (Gostar, Hoseinnezhad and Bab-Hadiashar, 2016; Wang et al., 2018) and using only a single look-ahead horizon. (Dames, Tokekar and Kumar, 2017) proposed to control multiple fixed-wing UAVs to localise mobile taxis with a single objective value function. For localising and searching objects simultaneously, (Dames and Kumar, 2015) and (Charrow, Michael and Kumar, 2015) considered a similar scenario, but only for stationary objects. Planning using multi-objective optimisation (MOP) has not been explored yet, except for single sensor selection (Zhu, Wang and Liang, 2019) or using the weighted sum method presented in (Charrow, Michael and Kumar, 2015) where the weighting parameters are difficult to define without prior knowledge. In contrast, we focus on optimising all value functions (*i.e.*, tracking and discovering) simultaneously using MOP. In particular, our proposed tracking and discovering value functions are based on information criteria. The tracking value function maximises the mutual information between future measurements and discovered object states under a multi-sensor Bernoulli filter; the discovering value function maximises the mutual information between empty measurements and undiscovered object states under a grid occupancy filter.

**Our contributions:** The main contributions of our work are: *(i)* We formulate a multi-agent planning problem with competing objectives and propose a planning algorithm for searching and tracking multiple mobile objects; *(ii)* We unify tracking and planning algorithms under a Bernoulli-based model; *(iii)* We prove that our proposed

multi-objective value function is submodular; hence, the greedy algorithm can be used to rapidly determine the approximately optimal control actions with a bounded performance guarantee at $(1-1/e)$OPT.

## 5.3 Problem Formulation

First, we introduce assumptions to help define our problem and introduce the notations we adopt in our work. Second, we provide a brief overview of the multi-sensor Bernoulli filter which unifies the tracking and discovering formulation. Next, we formulate our MPOMDP multi-agent planning approach for controlling the multi-agent team.

### 5.3.1 Assumptions and notations

We consider a team of $S$ agents surveying a large area to detect and track an unknown and time-varying number of mobile objects using detection-based measurements. We assume that each agent can localise itself (*e.g.*, using an onboard GPS for UAVs) and that all agents can communicate to a central node to enable us to adopt the centralised approach for MPOMDP. Consequently, we assume that all of the measurements are transferred to a central node that analyses received information and subsequently sends control actions to all of the agents. Here, we employ a discrete control action space to reduce the computational load (Beard et al., 2017; Dames, Tokekar and Kumar, 2017). We further assume that the measurements from an object collected by the agents are conditionally independent given the object's state (Thrun, Burgard and Fox, 2005; Charrow, Michael and Kumar, 2015).

We use the convention that lower-case letters (*e.g.*, $x$) represent single-object states, upper-case letters (*e.g.*, $X$) represent multi-object (finite-set) states, and blackboard bold letters (*e.g.*, $\mathbb{X}, \mathbb{Z}$) represent spaces. We denote the inner product $\int f(x)g(x)dx = \langle f, g \rangle$.

### 5.3.2 Multi-sensor Bernoulli filter (MS-BF)

In practice, an object can randomly enter and leave the surveillance region, hence the number of objects of interest is unknown and time-varying. Further, it is important to consider the existence of objects of interest to allow the agents to discover new

objects when they enter the scene and to prevent agents following false-positives. This can be addressed by the random finite sets (RFSs) approach, first proposed by (Mahler, 2007b). RFSs are finite-set valued random variables. We assume that each measurement is uniquely identified, *e.g.*, transmit frequencies from radio beacons (Kays et al., 2011; Thomas, Holland and Minot, 2012; Cliff et al., 2015; Nguyen et al., 2018b, 2019a,b) or MAC address (Beck et al., 2018; Charrow, Michael and Kumar, 2015), which is typical for wildlife tracking or search and rescue missions. Since each object is uniquely identified, we propose using a the multi-sensor Bernoulli filter (MS-BF) (Vo et al., 2012), where each object's state is a Bernoulli RFS, and run multiple MS-BF filters parallel to track multiple objects. A Bernoulli RFS $X$ on $\mathbb{X}$ has at most one element with probability $r$ for being singleton or $1-r$ for being empty. Its probability density $\pi(\cdot) = (r, p(\cdot))$ given by

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

**Object tracking with MS-BF:** We model each object's state at time $k$ by $X_k$ as a Bernoulli RFS. The MS-BF propagates the two quantities: the existence probability $r$ and spatial density $p(\cdot)$. If the posterior density is $\pi_{k-1} = (r_{k-1}, p_{k-1})$, then the predicted density $\pi_{k|k-1} = (r_{k|k-1}, p_{k|k-1})$ is also a Bernoulli RFS, with $r_{k|k-1} = r_{B,k}(1-r_{k-1}) + r_{k-1}\langle p_{S,k}, p_{k-1}\rangle$, $p_{k-1}(x_k) = \left[r_{B,k}(1-r_{k-1})b_k(x_k) + r_{k-1}\langle f_{k|k-1}(x_k|\cdot), p_{S,k}(\cdot)p_{k-1}(\cdot)\rangle\right]/r_{k|k-1}$. Here, $r_{B,k}$ and $p_{S,k}$ are the probabilities of object birth and object survival, $b_k(\cdot)$ is the object birth density. Further, the updated density $\pi_k$ is also a Bernoulli RFS, given by $\pi_k = (r_k, p_k)$ with $r_k = \Psi_k^{(S)} \circ \cdots \circ \Psi_k^{(1)}(r_{k|k-1})$; $p_k = \Psi_k^{(S)} \circ \cdots \circ \Psi_k^{(1)}(p_{k|k-1})$. Here, $\circ$ denotes composition (of operators), $\Psi_k^{(s)}$ is an update operator for agent $s$, *i.e.*:

$$[\Psi_k^{(s)}(r)] = \langle \eta^{(s)}(Z^{(s)}|\cdot), p(\cdot)\rangle r / \left[(1-r)e^{-\lambda^{(s)}} r \langle \eta^{(s)}(Z^{(s)}|\cdot), p(\cdot)\rangle\right], \tag{5.1}$$

$$[\Psi_k^{(s)}(p)](x) = \eta^{(s)}(Z^{(s)}|x)p(x) / \langle \eta^{(s)}(Z^{(s)}|\cdot), p(\cdot)\rangle, \tag{5.2}$$

where the superscript $(s)$ denotes the parameters of agent $s$, $\lambda^{(s)}$ is the clutter rate, and $\eta^{(s)}(Z^{(s)}|x)$ denotes the likelihood of measurement set $Z^{(s)}$ from agent $s$ given the object's state $x$. $\eta^{(s)}(Z^{(s)}|x)$ is also a Bernoulli RFS, given by

$$\eta^{(s)}(Z^{(s)}|x) = \begin{cases} 1 - p_d^{(s)}(x), & \text{if } Z^{(s)} = \emptyset, \\ p_d^{(s)}(x) g^{(s)}(z|x), & \text{if } Z^{(s)} = \{z\}. \end{cases} \tag{5.3}$$

Here, $p_d^{(s)}(x)$ is the probability that agent $s$ detects object $x$, and $g^{(s)}(z|x)$ is the (conventional) likelihood function of measurement $z$ given object's state $x$.

### 5.3.3 Planning

At time $k$, the team of $S$ agents needs to plan how they manoeuvre over the time interval $k+1:k+H$ to improve its estimation of the states of multiple objects $X_k$, where $H$ denotes the look-ahead horizon length. Let $\mathbb{A} \subseteq \mathbb{R}^N$ be all possible set of control actions for a given agent. When the control action $a_k^i \in \mathbb{A}$ is applied to an agent $i$, it follows a trajectory comprised of sequence of the discrete poses $u_{k+1:k+H}^i(a_k^i) = [u_{k+1}^i, \ldots, u_{k+H}^i]^T$ with corresponding measurements $Z_{k+1:k+H}^i(a_k^i) = [Z_{k+1}^i, \ldots, Z_{k+H}^i]^T$ (for notational compactness, we omit the dependence on $X_k$ here). Let $A_k = [a_k^1, \ldots, a_k^S]^T \in \mathbb{A}^S$ be the control actions where $\mathbb{A}^S = \mathbb{A} \times \cdots \times \mathbb{A}$ is the control action space for $S$ agents, and the corresponding measurement set is $Z_{k+1:k+H}(A_k) = [Z_{k+1:k+H}^1(a_k^1), \ldots, Z_{k+1:k+H}^S(a_k^S)]^T$.

The objective of path planning is to find the optimal action $A_k^* \in \mathbb{A}^S$ that maximises the value function, *i.e.*,

$$A_k^* = \underset{A_k \in \mathbb{A}^S}{\arg\max}\, V(X_{k+1:k+H}, Z_{k+1:k+H}(A_k)). \tag{5.4}$$

where $V(X_{k+1:k+H}, Z_{k+1:k+H}(A_k)) = \boldsymbol{E}\left[\sum_{j=1}^{H} \mathcal{R}(X_{k+j}, Z_{k+j}(A_k))\right]$ is the value function or the expected sum of immediate rewards $\mathcal{R}(\cdot)$ over a finite horizon $H$.

Since an analytic solution does not exist for the expected reward, we use the predicted ideal measurement set (PIMS) (Mahler, 2004)—a computationally low-cost approach. The value function is calculated:

$$V(X_{k+1:k+H}, \hat{Z}_{k+1:k+H}(A_k)) = \sum_{j=1}^{H} \mathcal{R}(X_{k+j}, \hat{Z}_{k+j}(A_k)), \tag{5.5}$$

where $\hat{Z}_{k+j}(A_k)$ denotes the ideal measurement set of $Z_{k+j}(A_k)$ calculated using the measurement model and the estimated states of objects without measurement noise. For notational compactness, we write the value function $V(X_{k+1:k+H}, \hat{Z}_{k+1:k+H}(A_k))$ as $V(A_k)$.

## 5.4 Planning for Tracking and Discovering Multiple Objects

### 5.4.1 Planning for tracking discovered mobile objects

In this problem, we consider maximising an information-based reward function to reduce the overall uncertainty of the discovered mobile objects because more information naturally implies less uncertainty. In particular, we propose using the mutual information $I(X;Z)$ between the object's state $X$ and measurement state $Z$ as the immediate reward function, and the long-term sum of rewards over a finite horizon $H$, or so-called the value function is given by

$$V_1(A_k) = \sum_{j=1}^{H} I(X_{k+j}; \hat{Z}_{k+j}(A_k)), \tag{5.6}$$

where $I(X;Z) = h(X) - h(X|Z)$, with $h(X)$ is the generalisation of differential entropy for a finite set $X \subseteq \mathbb{X}$ with density $f(X)$ defined as $h(X) = -\int_{\mathbb{X}} f(X) \log f(X) \delta X$; here $\int_{\mathbb{X}} \cdot \delta X$ is the set integral (Mahler, 2007b). For Bernoulli RFS, this integration is simplified to $h(X) = -\left[f(X = \emptyset) \log f(X = \emptyset) + \int f(X = x) \log f(X = x) dx\right]$. We have the following theorem.

**Theorem 5.1.** *The mutual information $I(X;Z)$ between the object state $X$ and measurement state $Z$ is a monotone submodular set function of $Z$.*

*Proof.* We want to prove that this mutual information $I(X;Z)$ is a monotone submodular set function, *i.e.*, for $Z_1 \subseteq Z_2 \subseteq \mathbb{Z}$, and $z \in \mathbb{Z} \setminus Z_2$ independent of $Z_1$ and $Z_2$:

$$I(X; Z_2, \{z\}) - I(X; Z_2) \leq I(X; Z_1, \{z\}) - I(X; Z_1).$$

Since $Z_1 \subseteq Z_2 \subseteq \mathbb{Z}$, using mutual information inequalities (Cover and Thomas, 2012, pp.50), we have:

$$\begin{aligned} & I(Z_2; \{z\}) \geq I(Z_1; \{z\}), \\ \Leftrightarrow\ & h(z) - h(z|Z_2) \geq h(z) - h(z|Z_1), \\ \Leftrightarrow\ & h(z|Z_1) \geq h(z|Z_2), \\ \Leftrightarrow\ & h(Z_1, \{z\}) - h(Z_1) \geq h(Z_2, \{z\}) - h(Z_2). \end{aligned} \tag{5.7}$$

Further, since $I(Z_2;\{z\}|X) = I(Z_1;\{z\}|X) = 0$ is due to $z$ is independent of $Z_1$ and $Z_2$ given $X$, we have:

$$\begin{aligned} h(\{z\}|X) &= h(\{z\}|X,Z_2) + I(Z_2;\{z\}|X) = h(\{z\}|X,Z_2) \\ &= h(X,Z_2,\{z\}) - h(X,Z_2), \\ h(\{z\}|X) &= h(\{z\}|X,Z_1) + I(Z_1;\{z\}|X) \\ &= h(X,Z_1,\{z\}) - h(X,Z_1). \end{aligned}$$

Hence,

$$h(X,Z_2,\{z\}) - h(X,Z_2) = h(X,Z_1,\{z\}) - h(X,Z_1). \tag{5.8}$$

Subtracting (5.7) from (5.8), we have:

$$\begin{aligned} &[h(X,Z_2,\{z\}) - h(X,Z_2)] - [h(Z_2,\{z\}) - h(Z_2)] \\ &\geq [h(X,Z_1,\{z\}) - h(X,Z_1)] - [h(Z_1,\{z\}) - h(Z_1)]. \end{aligned}$$

Using differential entropy chain rules (Cover and Thomas, 2012, pp.253), we have that $h(X|Z_2,\{z\}) = h(X,Z_2,\{z\}) - h(Z_2,\{z\})$ and $h(X|Z_2) = h(X,Z_2) - h(Z_2)$, thus the above equation is equivalent to

$$\begin{aligned} &h(X|Z_2,\{z\}) - h(X|Z_2) \geq h(X|Z_1,\{z\}) - h(X|Z_1) \\ \Leftrightarrow &[h(X) - h(X|Z_2,\{z\})] - [h(X) - h(X|Z_2)] \\ &\leq [h(X) - h(X|Z_1,\{z\}] - [h(X) - h(X|Z_1)], \\ \Leftrightarrow &I(X;Z_2,\{z\}) - I(X;Z_2) \leq I(X;Z_1,\{z\}) - I(X;Z_1). \end{aligned}$$

Thus, $I(X;Z)$ is a submodular set function. Further, using the chain rule we have:

$$I(X;Z_2,\{z\}) - I(X;Z_2) = I(X;Z_2|\{z\}) \geq 0. \tag{5.9}$$

Therefore, $I(X;Z)$ is a monotone submodular set function. □

**Remark:** Our mutual information formulation is different to that in (Krause, Singh and Guestrin, 2008) used for sensor selection problems. Krause et al. showed that for $Z \subseteq \mathbb{Z}$, the mutual information $I(Z;\mathbb{Z} \setminus Z)$ is a submodular set function. In other words, the mutual information $I(Z_1;Z_2)$ is submodular with the *property* that $Z_1 \cup Z_2 = \mathbb{Z}$ and $|\mathbb{Z}|$ is fixed. In contrast, we measure the mutual information between the random set object state $X$ and the random set measurement state $Z$ and prove $I(X;Z)$ is also a submodular set function of $Z$ without the aforementioned *property*.

**Corollary 5.2.** *The value function $V_1(A_k)$ in (5.6) is a monotone submodular set function.*

*Proof.* Since $I(X_k; \hat{Z}_{k+j}(A_k))$ is a monotone submodular set function and $V_1(A_k)$ is a positive linear combination of it, according to (Nemhauser, Wolsey and Fisher, 1978, pp.272), $V_1(A_k)$ is a monotone submodular set function. □

**Mutual Information Calculation based on MS-BF**: Assume that each object $i$ is associated with a Bernoulli distribution $\pi(X_i) = (r_i, p_i)$. Let $p_i(x)$ be approximated by a set of $N_s$ particles, such that $p_i(x) \approx \sum_{m=1}^{N_s} w_i^{(m)} \delta(x^{(m)} - x)$ with $\sum_{m=1}^{N_s} w_i^{(m)} = 1$ and $\delta(\cdot)$ is the Kronecker delta function, $X = X_1 \cup \cdots \cup X_n$ be the state of multiple objects. Since each object is uniquely identified by its label and estimated by an individual Bernoulli filter, we have

$$h(X) = \sum_{i}^{n} h(X_i) \approx \sum_{i}^{n} \Big[ -(1 - r_i)\log(1 - r_i) - r_i \sum_{m=1}^{N_s} \big[ w_i^{(m)} \log(r_o w_i^{(m)}) \big] \Big]. \tag{5.10}$$

According to the definition of the mutual information $I(X;Z) = h(X) - h(X|Z)$, thus the tracking value function $V_1(A_k)$ can be calculated as $V_1(A_k) = \sum_{j=1}^{H} \big[ h(X_{k+j}) - h(X_{k+j}|\hat{Z}_{k+j}(A_k)) \big]$, where $h(X_{k+1})$ is calculated directly in (5.10). For $h(X_{k+j}|\hat{Z}_{k+j}(A_k))$, it has the same form as in (5.10); however, $r_{k+j,i}$ and $w_{k+j,i}^{(m)}$ are calculated by propagating $r_{k,i}$ and $w_{k,i}^{(m)}$ from time $k$ to $k+j$ using (5.1) and (5.2) respectively with the ideal measurements $\hat{Z}_{k+j}(A_k)$.

### 5.4.2 Planning to search for undiscovered mobile objects

**Occupancy Grid Filter:** Since an agent is equipped with a sensor with a limited detection range, we propose using an occupancy grid to represent the probability of any undiscovered objects (Elfes, 1989). We extend the static grid approach in (Charrow, Michael and Kumar, 2015; Thrun, Burgard and Fox, 2005) by incorporating the birth probability into each occupancy cell to account for the possibilities of mobile objects entering and leaving the survey area, anytime. The survey area is divided into an occupancy grid $G = \{g^1, \ldots, g^{N_g}\} \subset \mathbb{R}^N$, where each cell $g^i \in G$ is associated with a Bernoulli random variable $r^i$. Here, $r^i$ is the probability that cell $g^i$ contains at least one undiscovered object. For initialisation, we set $r_0^i = r_B$ such that every cell has the same prior. Each cell $i$ is propagated through MS-BF over time using the predict

and update equations. In particular, let $r^i_{k-1}$ be the probability of cell $g^i$ containing at least one undiscovered object, then its predict and update probabilities at time $k$ are (5.11) and (5.12). Note that since these objects are yet to be discovered, we use empty measurements for all agents (denoted as $Z^{\emptyset}$ to update).

$$r^i_{k|k-1} = r_B(1 - r^i_{k-1}) + r^i_{k-1} p_S, \tag{5.11}$$

$$r^i_k = \Psi^{(S)}_k \circ \cdots \circ \Psi^{(1)}_k (r^i_{k|k-1}), \tag{5.12}$$

where $[\Psi^{(s)}_k(r^i)] = (1 - p^{(s)}_d(g^i))r^i / [1 - r^i + r^i(1 - p^{(s)}_d(g^i))]$.

**Searching for undiscovered objects:** As before, we propose using mutual information as the immediate reward function. We want to maximise the mutual information between the estimated occupancy grid $G$ and the ideal empty future measurement $\hat{Z}^{\emptyset}_{k+1:k+H}(A_k)$, *i.e.*,

$$V_2(A_k) = \sum_{j=1}^{H} I(G_k; \hat{Z}^{\emptyset}_{k+j}(A_k)), \tag{5.13}$$

where $I(G_k; \hat{Z}^{\emptyset}_{k+j}(A_k)) = \mathcal{H}(G_k) - \mathcal{H}(G_k|\hat{Z}^{\emptyset}_{k+j}(A_k))$ and $\mathcal{H}(G_k)$ is the Shannon entropy of $G_k$:

$$\mathcal{H}(G_k) = -\sum_{i=1}^{N_g} \left[ r^i_k \log(r^i_k) + (1 - r^i_k) \log(1 - r^i_k) \right], \tag{5.14}$$

and $\mathcal{H}(G_{k+j}|\hat{Z}^{\emptyset}_{k+j}(A_k))$ has the same form as in (5.14) with $r^i_{k+j}$ is calculated by propagating $r^i_{k+j}$ from $k$ to $k+j$ using the update step in (5.12) with empty measurements $\hat{Z}^{\emptyset}_{k+j}(A_k)$.

**Theorem 5.3.** *The value function $V_2$ in (5.13) is a monotone submodular set function.*

*Proof.* We can apply a similar strategy as per Theorem 5.1 to prove that $I(G_k; \hat{Z}^{\emptyset}_{k+j}(A_k))$ is a monotone submodular set function, note that $\mathcal{H}(\cdot)$ is the Shannon entropy (a discrete version of differential entropy $h(\cdot)$). Further, since $V_2(A_k)$ is a positive linear combination of $I(G_k; \hat{Z}^{\emptyset}_{k+j}(A_k))$, according to (Nemhauser, Wolsey and Fisher, 1978, pp.272), $V_2(A_k)$ is a monotone submodular set function. □

### 5.4.3 Multi-objective value function for tracking and discovering

In this problem, we want to control the team of agents to perform both tracking and discovering; this naturally leads to a multi-objective problem. Specifically, we want to

maximise

$$V(A_k) = [V_1(A_k), V_2(A_k)]^T, \tag{5.15}$$

subject to $A_k \in \mathbb{A}^S$ where $V_1$ and $V_2$ are defined in (5.6) and (5.13), respectively. Multi-objective optimisation provides a meaningful notion of multi-objective optimality such as the Pareto-set, which represents trade-offs between the objectives such that there is no other solution that can improve one objective without degrading any remaining objectives (Whiteson and Roijers, 2016). Online planning necessitates selecting one compromised solution from the Pareto-set on-the-fly. One approach is robust submodular observation selection (ROSS) (Krause et al., 2008), which is robust against the worst possible objective; however, even if each $V_i$ is submodular, $V_{min} = \min_i V_i$ is generally not submodular. Other approaches include weighted sum (WS) and global criterion method (GCM); the simplicity of these methods are not only attractive for meeting the demands of online planning but also result in a submodular value function. In this work, we adopt GCM to select the compromised solution considering the distance equally for two value functions from the ideal solution. Inspired by (Koski, 1993), we define the value function $V_{mo}$ (with $V_{mo}(\emptyset) = 0$) as:

$$V_{mo}(A_k) = \sum_{i=1}^{2} \frac{V_i(A_k) - \min\limits_{A_k \in \mathbb{A}^S} V_i(A_k)}{\max\limits_{A \in \mathbb{A}^S} V_i(A_k) - \min\limits_{A_k \in \mathbb{A}^S} V_i(A_k)}. \tag{5.16}$$

The global criterion method admits a unique optimal solution from the Pareto-set (Coello et al., 2007). Hence, the multi-objective problem becomes

$$A_k^* = \arg\max_{A_k \in \mathbb{A}^S} V_{mo}(A_k). \tag{5.17}$$

Since finding the optimal control action $A^* \in \mathbb{A}^S$ is a combinatorial optimisation problem, we want to show that the multi-objective value function $V_{mo}(A)$ in (5.16) is also a monotone submodular set function on $Z$. This enables us to use the greedy algorithm to find the optimal action that approximately maximise this multi-objective value function.

**Corollary 5.4.** *The multi-objective value function $V_{mo}$ in (5.16) is a monotone submodular set function.*

*Proof.* Since $V_i(A_k))$ is a monotone submodular set function and $V_{mo}(A_k)$ is a positive linear combination of it, according to (Nemhauser, Wolsey and Fisher, 1978, pp.272), $V_{mo}(A_k)$ is a monotone submodular set function. □

### 5.4.4 Greedy search algorithm

We proved that our multi-objective value function $V_{mo}(\cdot)$ is a monotone submodular set function—see **Corollary 4**. For submodular functions, (Nemhauser, Wolsey and Fisher, 1978) proved the greedy search algorithm guarantees a performance bound at $(1-1/e)$OPT, where OPT is the optimal value of the submodular function. Therefore, if the optimal value of our value function is $V_{mo}(A^*)$, we can simply state the following fundamental performance bound for our submodular value function:

**Theorem 5.5.** *From (Nemhauser, Wolsey and Fisher, 1978). Let $A_G$ be the output greedy control action and $A^*$ be the optimal control action evaluated using brute-force method of (5.17). Then*

$$V_{mo}(A_G) \geq (1-1/e)V_{mo}(A^*), \tag{5.18}$$

*where $e = 2.718\ldots$ is the base of the natural logarithm.*

Hence, we propose using the greedy search algorithm by simply adding agents sequentially and picking the next agent which provides the maximum value function $V_{mo}(\cdot)$ as presented in Algorithm 5.1.

**Algorithm 5.1** Greedy algorithm

**Input**: $V_{mo}(\cdot), \mathbb{A}$ ▷ value function and the action space.
**Output**: $A_G \in \mathbb{A}^S$ ▷ greedy control actions for all agents.
$A_G := \emptyset$ ▷ initialise the greedy control action.
$P := \emptyset$ ▷ initialise the agent planned list.
$U := \{1, \ldots, S\}$ ▷ initialise list of agents to plan.
**while** $U \neq \emptyset$ **do**
    **for each** $s \in U$ **do**
        $[A^s, V_c^s] := \underset{A \in \mathbb{A}^{V \cup \{s\}}}{\arg\max}\, V_{mo}(A)$ ▷ find the best action and value function for each agent in $U$.
    **end for**
    $s^* := \underset{s \in U}{\arg\max}\, V_c^s$ ▷ select the agent $s^*$ that provides the best value function.
    $A_G := A_G \cup \{A^{s^*}\}$ ▷ save the greedy control action for agent $s^*$.
    $P := P \cup \{s^*\}$ ▷ add agent $s^*$ into the planned list.
    $U := U \setminus \{s^*\}$ ▷ remove agent $s^*$ from the list of agents to plan.
**end while**

## 5.5 Experiments

We evaluate the proposed value function using a series of comprehensive synthetic experiments since we can control all of the parameters of the problem, especially with a time-varying number of agents and objects. We compare three planning algorithm formulations: ***(i)*** using the single objective value function $V_1(\cdot)$ in (5.6) for tracking. ***(ii)*** using a single objective value function based on our *new* discovery value function $V_2(\cdot)$ in (5.13). ***(iii)*** using our proposed multi-objective value function $V_{mo}(\cdot)$.

We use optimal sub-pattern assignment (OSPA) (Schuhmacher, Vo and Vo, 2008) to measure performance. We report **OSPA Dist** as the ***main*** metric to evaluate the *overall* team performance since it incorporates both tracking and discovery indicators. For further insights into our planning formulations, we also report: ***(i)*** **OSPA Loc** as a localisation accuracy measure, ***(ii)*** **OSPA Card** as an object discovery performance measure, and ***(iii)*** **Search Area Entropy** as the average entropy of the occupancy grid to measure the coverage area of the team. For demonstration, a team of quad-copter UAVs flying at different altitudes is considered. The detailed parameter settings are provided as follows, while scenario setups are shown in Figure 5.1. Our experiments considered for different scenarios and two different detection-based sensors subject to noisy measurements.

**Parameter settings for experiments:** The search areas for the first three scenarios and scenario 4 are 1000 m $\times$1000 m and 2000 m $\times$2000 m, respectively. Each agent is controlled to fly at a fixed and different altitude (*i.e.*, 5 m altitude gap between each agent) to prevent collisions with other team members. The minimum altitude starts at 30 m for the first agent and increases 5 m for each additional agent. Further, all objects are assumed to exist on a horizontal ground plane to speed up the numerical experiments by tracking in 2D. Each object state $\boldsymbol{x} = (x, l)$ is uniquely identified by its label $l$, while its motion state $x = [p_x, \dot{p}_x, p_y, \dot{p}_y]^T$ comprises of object's position and velocity in Cartesian coordinates. Each object moves in accordance with the constant velocity (CV) model given by $x_k = F^{CV} x_{k-1} + q_{k-1}^{CV}$. Here, $F^{CV} = [1, T_0; 0, T_0] \otimes I_2$, $T_0$ is the sampling interval ($T_0 = 1$ s for our experiments), $\otimes$ denotes for the Kronecker tensor product; $I_2$ is the $2 \times 2$ identity matrix; $q_{k-1}^{CV} \sim \mathcal{N}(0, Q^{CV})$ is a $4 \times 1$ zero mean Gaussian process noise, with co-variance $Q^{CV} = \sigma_{CV}^2 [T_0^3/3, T_0^2/2; T_0^2/2, T_0] \otimes I_2$. The

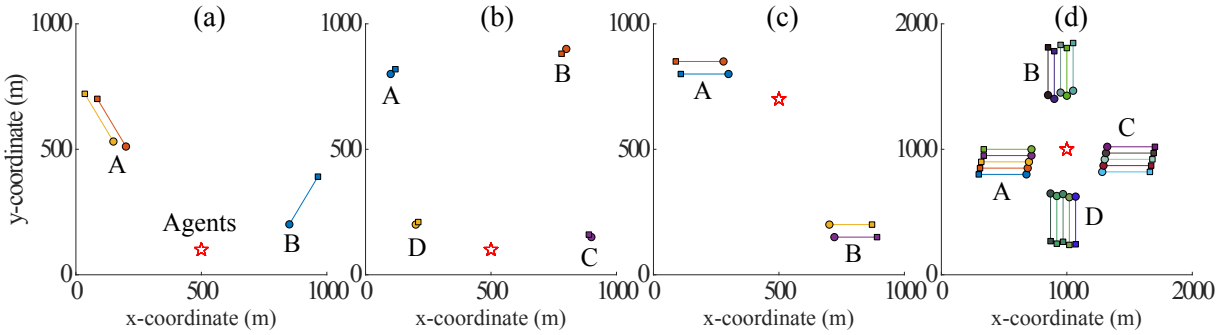


**Figure 5.1. Setup for four scenarios.** (a) Scenario 1; (b) Scenario 2; (c) Scenario 3; (d) Scenario 4. Start/Stop locations for each object are denoted by ◯/□. Start locations for agents are denoted by ⋆.

detection probability is

$$p_D(u_s, x_p) = \begin{cases} 0.98 & \text{if } ||x_p - u^s|| \leq r_d, \\ \max(0, 0.98 - (||x_p - u^s|| - r_d)\hbar) & \text{otherwise,} \end{cases}$$

where $r_d$ is the sensor detection range and $\hbar = 0.008$ m$^{-1}$. The sensor reports *false-detection* or false-alarm measurements following a Poison RFS with a clutter rate $\lambda = 0.2$, where each agent collects at most one measurement per time step for each object, either from the real objects, clutters (false-detection) or the measurement is empty (missed detection). For the sensor noise, the *range and bearing*-based measurement is corrupted by zero-mean Gaussian noise that depends on the distance between objects and agents, *i.e.*, $v \sim \mathcal{N}(0, R)$ with $R = \text{diag}(\sigma_\phi^2, \sigma_\rho^2)$ where $\sigma_\phi = \sigma_{0,\phi} + \beta_\phi ||x_p - u^s||$, $\sigma_\rho = \sigma_{0,\rho} + \beta_\rho ||x_p - u^s||$, $\sigma_{0,\phi} = 2\pi/180$ rad, $\beta_\phi = 1.7 \cdot 10^{-5}$ rad/m, $\sigma_{0,\rho} = 10$ m, and $\beta_\rho = 5 \cdot 10^{-3}$. Similarly, for *vision-based* sensor, each detected object $x$ leads to a measurement $z$ of noisy $x - y$ positions, given by: $z = [p_x, p_y]^T + v$. Here, $v \sim \mathcal{N}(0, R)$ with $R = \text{diag}(\sigma_x^2, \sigma_y^2)$ where $\sigma_x = \sigma_y = \sigma_{0,xy} + \beta_{xy} ||x_p - u^s||$ with $\sigma_{0,xy} = 10$ m, and $\beta_{xy} = 1 \cdot 10^{-2}$. The grid size is $100 \times 100$ across four scenarios. This corresponds to a grid cell of 10 m $\times$10 m for scenario 1,2 and 3 and a grid cell of 20 m $\times$20 m for scenario 4. The total time is 200 s. The agent does not have any prior knowledge about object's state, thus it uses the initial birth probability $r_B = 0.005$, and a Gaussian density $p_B = \mathcal{N}(x; m_B, Q_B)$ with $m_B = [500, 0, 500, 0]^T$ and $Q_B = \text{diag}([500, 10, 500, 10])$.

**Scenario 1 (FastMoving)**: *Three fast moving objects in two groups travelling in the same direction.* A team of agents starts at $[500, 100]^T$ m as depicted in Figure 5.1a.

**Scenario 2 (LateBirth)**: *Late birth objects*. We investigate a searching and tracking scenario in Figure 5.1b) with *four* slow-moving mobile objects using a team of agents. Here, the groups of objects $D$ and $C$ enter the scene when the agents are out of their detection range—late birth. This scenario *favours* agent planning with the discovery value function encouraging exploration and demonstrates the effectiveness of our multi-objective value function with its competing tracking and discovery objectives.

**Scenario 3 (Opposite)**: *Four objects in two groups (A and B) moving rapidly in opposing directions*. Figure 5.1c illustrates the scenario. We use this setting to confirm the effectiveness of our multi-objective value function. Now, the possibility to discover group $B$ out of the sensor detection range must be achieved through exploration while planning to track group $A$ in the vicinity of the agents is immediately rewarded by the tracking objective.

**Scenario 4 (Explosion)**: *Multiple groups of fast moving objects in opposing directions*. Here, we consider a team of agents to search and track *20 fast moving mobile objects* as shown in Figure 5.1d.

**Detection-based sensors:** ***(i)*** We considered agents equipped with a range and bearing based sensor—common in wildlife tracking (Cliff et al., 2015) for example. Let $u^s = [p_x^s, p_y^s, p_z^s]^T$ be the position of agent $s$, $x_p = [p_x, p_y, 1]^T$ be the position of object $x$, each detected object $x$ leads to a noisy measurement $z$ of range and bearing given by: $z = \left[\arctan\left[(p_y - p_y^s)/(p_x - p_x^s)\right], ||x_p - u^s||\right]^T + v$. Here,$||\cdot||$ is the Euclidean norm; $v \sim \mathcal{N}(0, R)$ with $R = \text{diag}(\sigma_\phi^2, \sigma_\rho^2)$ where $\sigma_\phi = \sigma_{0,\phi} + \beta_\phi ||x_p - u^s||$, $\sigma_\rho = \sigma_{0,\rho} + \beta_\rho ||x_p - u^s||$. ***(ii)*** To demonstrate the sensor-agnostic nature of our approach, we consider agents equipped with a vision-based sensor. Each detected object $x$ leads to a measurement $z$ of noisy $xy$ positions, given by: $z = [p_x, p_y]^T + v$. Here, $v \sim \mathcal{N}(0, R)$ with $R = \text{diag}(\sigma_x^2, \sigma_y^2)$.

### 5.5.1 Comparing greedy and brute force algorithm results for our submodular multi-objective value function

Figure 5.2 depicts the ratio of our multi-objective value function obtained from greedy and brute-force algorithms for the four scenarios. The result obtained from 20 Monte Carlo (MC) runs for each scenario agrees with the performance guarantee of the greedy algorithm to yield an approximately optimal solution with a bounded performance guarantee at $(1 - 1/e)$ OPT.

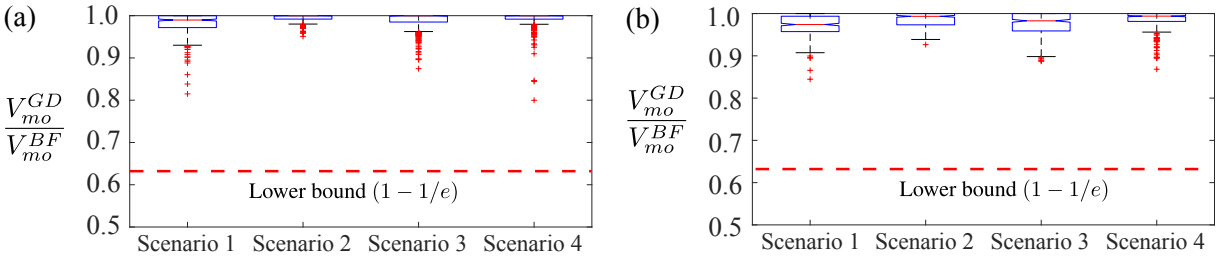


**Figure 5.2. Performance comparison between Greedy and Brute-force algorithms.** Multi-objective value function ratio between the greedy $V_{mo}^{GD}$ and brute-force $V_{mo}^{BF}$ algorithms with agents (a) $S = 2$ and (b) $S = 3$ (20 MC runs, range and bearing based sensor with $r_d = 200$ m).

### 5.5.2 Comparing multi-objective multi-agent planning with single objective multi-agent planning

Table 5.1 compares results for scenario 1, 2, 3 and 4 collected from 20 MC runs for agents with *range and bearing* based sensors. It is expected that the average search area entropy is smallest for $V_2$ since it encourages agents to explore the search area. Consequently, $V_2$ can also be seen to generate the best performance in terms of OSPA cardinality—**OSPA Card**. In contrast, we can see that the multi-agent planning with the single value function (to encourage only tracking accuracy) $V_1$, achieves improved results for object localisation accuracy only (low **OSPA Loc** results) but at the expense of missing objects often out of the range of the sensors (as seen by significantly large **OSPA Card** results). Most importantly, our results verify that $V_{mo}$ performs best in terms of overall tracking and cardinality accuracy (reported by **OSPA Dist**) since $V_{mo}$ not only rewards agents for undertaking the discovery of new objects but also rewards agents for accurately tracking discovered objects.

Figure 5.3 shows the grid occupancy probability and the trajectories of the agents for scenario 3. The results demonstrate the effectiveness of our proposed planning method, where agents not only track but discover distant mobile objects.

### 5.5.3 Explore the asymptotic behaviour of tracking performance with an increasing number of agents for our planning formulation

Figure 5.4 depicts the overall mean tracking accuracy from 20 MC runs for agent teams with each detection-based sensor. It confirms that planning with $V_{mo}$ consistently performs better than $V_1$ or $V_2$ alone. As expected, when the number of agents increases,

**Table 5.1. Comparing multi-agent planning for tracking mobile objects using our multi-objective value function $V_{mo}$ across four scenarios using range and bearing based sensors with detection range $r_d = 200$ m.** We use planning with the single objective value functions $V_1$ and $V_2$ as baselines and the results are averaged over 20 Monte Carlo experiments.

| | | Scenario 1 (FastMoving) | | | | Scenario 2 (LateBirth) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Indicators | Overall Performance | Tracking Performance | Discovery Performance | | Overall Performance | Tracking Performance | Discovery Performance | |
| *Agents* | *Value Functions* | **OSPA Dist (m)** | **OSPA Loc (m)** | **OSPA Card (m)** | **Search Area Entropy (nats)** | **OSPA Dist (m)** | **OSPA Loc (m)** | **OSPA Card (m)** | **Search Area Entropy (nats)** |
| $S = 3$ | $V_1$ | 33.9 | 4.4 | 29.5 | 0.23 | 57.0 | 4.0 | 53.0 | 0.22 |
| | $V_2$ | 21.2 | 9.7 | 11.5 | 0.12 | **41.1** | 10.3 | 30.8 | 0.12 |
| | $V_{mo}$ | **17.7** | 6.1 | 11.6 | 0.17 | 52.1 | 5.2 | 46.9 | 0.17 |
| $S = 5$ | $V_1$ | 25.4 | 5.1 | 20.3 | 0.20 | 53.4 | 3.6 | 49.8 | 0.17 |
| | $V_2$ | 20.3 | 9.2 | 11.1 | 0.09 | 43.9 | 9.5 | 34.4 | 0.09 |
| | $V_{mo}$ | **16.8** | 5.7 | 11.1 | 0.13 | **38.8** | 5.1 | 33.7 | 0.11 |
| | | Scenario 3 (Opposite) | | | | Scenario 4 (Explosion) | | | |
| $S = 3$ | $V_1$ | 51.7 | 3.0 | 48.7 | 0.24 | 53.7 | 3.1 | 50.6 | 0.32 |
| | $V_2$ | 18.3 | 12.8 | 5.5 | 0.12 | 55.0 | 29.8 | 25.2 | 0.28 |
| | $V_{mo}$ | **11.1** | 5.9 | 5.2 | 0.18 | **40.7** | 9.7 | 31.0 | 0.32 |
| $S = 5$ | $V_1$ | 51.2 | 2.9 | 48.3 | 0.24 | 36.9 | 5.0 | 31.9 | 0.30 |
| | $V_2$ | **10.5** | 6.9 | 3.6 | 0.09 | 31.2 | 19.5 | 11.7 | 0.25 |
| | $V_{mo}$ | **10.5** | 5.9 | 4.1 | 0.15 | **17.4** | 6.4 | 11.1 | 0.29 |

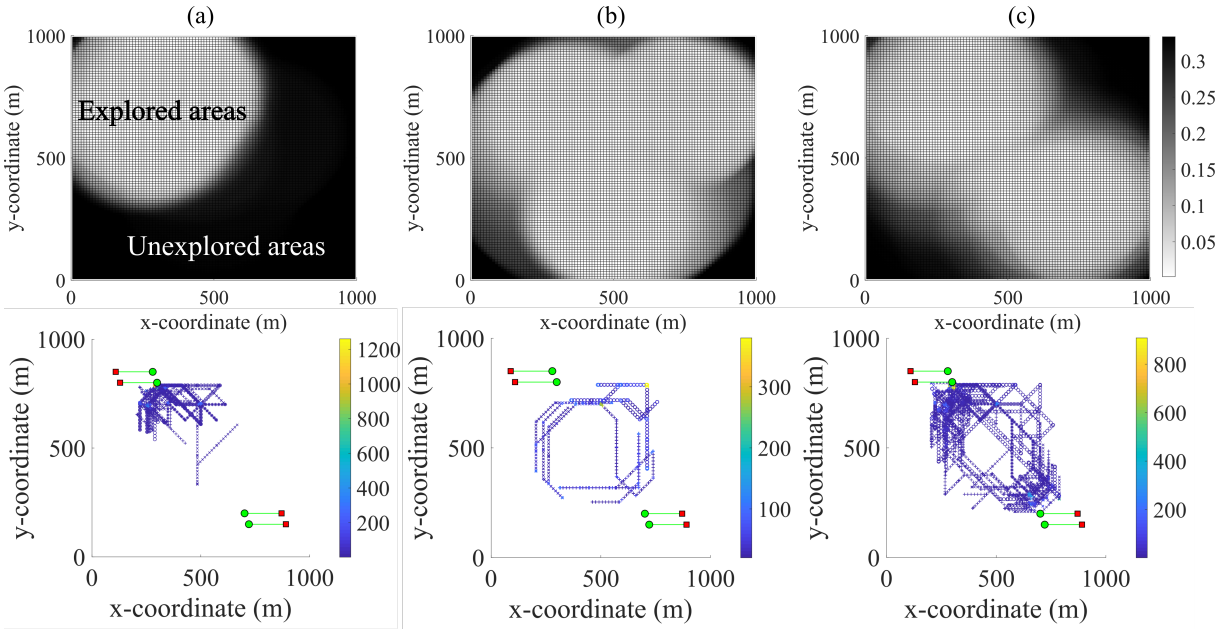


**Figure 5.3. Illustration of grid occupancy probability and trajectories' heat map for Scenario 3.** Grid occupancy probability (top) and heat map (bottom) of trajectories for 3 agents over 20 MC runs with $r_d = 200$ m using (a) $V_1$. Late birth group $B$ never discovered, (b) $V_2$. Extensive exploration, and (c) $V_{mo}$. Discovers the late birth group $B$ whilst tracking both groups.

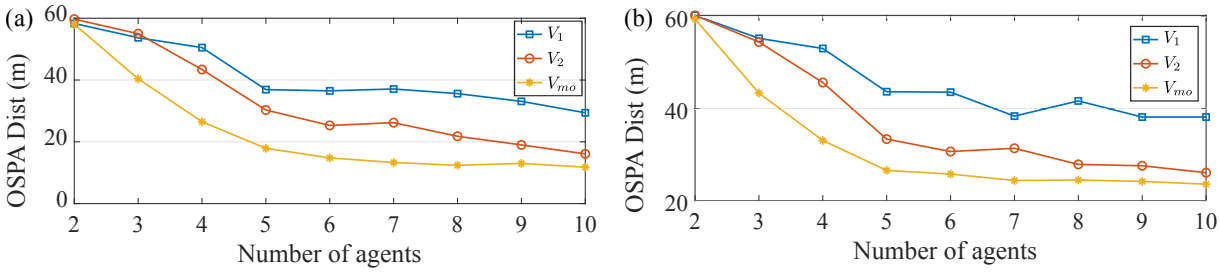


**Figure 5.4. Comparison between our multi-objective value function $V_{mo}$ versus baseline methods.** Overall tracking performance over 20 MC runs based on multi-agent planning with our multi-objective value function $V_{mo}$ compare with the single objective value functions $V_1$ and $V_2$ when the number of agents are increased from 2 to 10 for Scenario 4 (**Explosion**) with $r_d = 200$ m using (a) agents with *range and bearing* based sensors, (b) agents with *vision* based sensors.

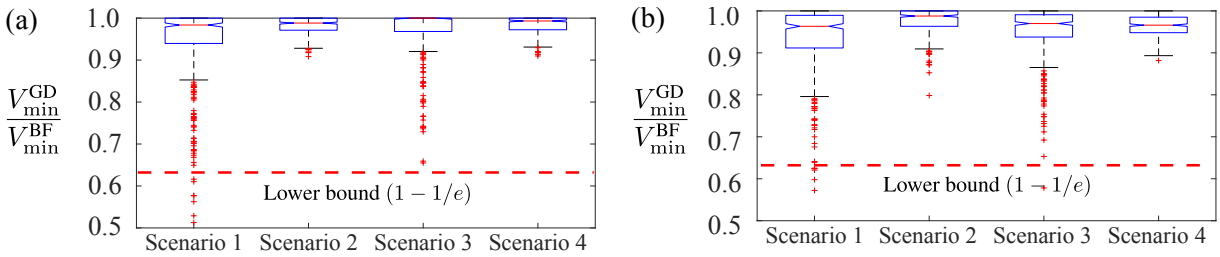


**Figure 5.5. Ratio of the RSOS-based value function $V_{\text{min}}$ between the greedy and brute-force algorithms over 20 MC-runs across four scenarios with the detection range $r_d = 200$ m.** (a) Agents $S = 2$ and (b) Agents $S = 3$. Here, $V^{GD}_{\text{min}}$ and $V^{BF}_{\text{min}}$ are the RSOS-based value functions evaluated with the greedy and brute-force algorithms.

$V_1$ and $V_2$ tracking performances improve and approach that of $V_{mo}$. Interestingly, multi-agent planning with a single exploration objective closely approaches the tracking performance of the multi-objective value function when the team of agents is large enough to cover the survey area with its range limited sensors and all objects become visible to the agents.

### 5.5.4 The evolution of the computational time with the number of agents

In this subsection, we investigate how the planning time evolves with respected to the number of agents using the proposed algorithm in Algorithm 5.1. Fig 5.6 depicts the planning time for a centralised controller to compute optimal control actions when

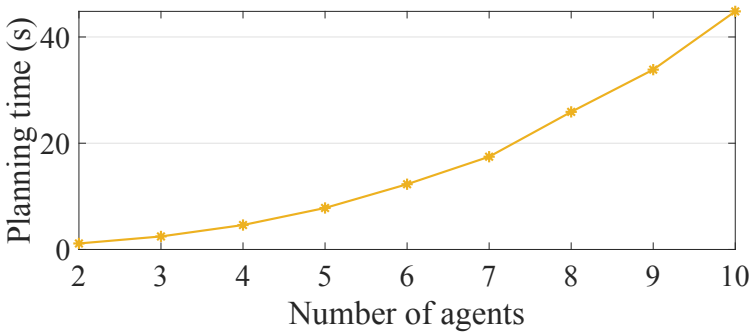


**Figure 5.6. The evolution of the computational time with respect to the number of agents.** The number of agents is increased from 2 to 10 for **Scenario 1 (FastMoving)** with $r_d = 200$ m using range and bearing-based sensors.

the number of agents $S$ is increased from 2 to 10 for **Scenario 1 (FastMoving)**. The results confirm that using the proposed algorithm, the planning time has quadratic complexity in the number of agents, as shown in Algorithm 5.1. In particular, the computational complexity of Algorithm 5.1 consists of two loops:

i) Computation of $V_{mo}$: $\mathcal{O}(V_{mo}) = \mathcal{O}(V_1) + \mathcal{O}(V_2) = \mathcal{O}(HN_s|X|S) + \mathcal{O}(HN_gS)$,

ii) Computation of $A_G$ using the greedy algorithm: $\mathcal{O}(S|A|)$.

Therefore, the total planning time complexity using the greedy algorithm is $\mathcal{O}\big(H(N_s|X| + N_g)|\mathbb{A}|S^2\big)$. Notably, using the greedy algorithm, the computational complexity of our proposed algorithm increases linearly with respect to the number of control actions $|\mathbb{A}|$ instead of increasing exponentially ($|\mathbb{A}|^S$) as in the case of the brute-force algorithm.

### 5.5.5 Comparing between greedy and brute force algorithms for a non-submodular multi-objective value function

We formulated and *proved* the submodularity of our multi-objective value function. Here, we investigate the non-submodular multi-objective optimisation method called robust submodular observation selection (RSOS) (Krause et al., 2008; Udwani, 2018), which is an a priori preference articulation method in multi-objective optimisation robust against the worst possible objective to contrast against the quality of the solutions from our multi-objective optimisation method based on the global criterion method. Unlike the global criterion method, where we *prove* the solution preference

formulation (see(5.16)) is submodular (see **Theorem** 5.5), in the RSOS approach, even if each of the $i^{\text{th}}$ value function $V_i$ is submodular (as in our formulations and *proofs*) , $V_{min} = \min_i V_i$ is generally not submodular (Krause et al., 2008).

Figure 5.5 depicts the ratio of the value function obtained from the RSOS multi-objective optimisation using greedy and brute-force algorithms for the four scenarios. The results obtained from 20 Monte Carlo (MC) runs for each scenario confirms the non-submodularity of the RSOS-based value function formulation since its performance using the greedy algorithm violates the $(1 - 1/e)$OPT (optimal) bound for scenario 1 and 3.

### 5.5.6 Grid occupancy probability (area coverage) and the trajectories of the agents

We illustrated the behaviour of the agents for Scenario 3 (**Opposite**) Section 5.5 as it provides an interesting mix of tracking and discovery. Here we present in Figures 5.7, 5.8, and 5.9 the grid occupancy probability and trajectory heat-maps resulting from the experiments for Scenario 1, 2 and 4 over 20 MC runs using a team of three agents.

As expected, value function $V_{mo}$ is not only able to track discovered mobile objects but also able to search for and track undiscovered mobile objects out of the range of the team's sensor detection range. This is evident in comparing the similarity between coverage plots for our discovery value function $V_2$ based planning method with that of our multi-objective value function $V_{mo}$ based planning. Performance of our multi-objective planning method is even more apparent in Figure 5.9 where the three agents fail to often visit all four corners of the search area when making planning decisions with only our discovery value function. In contrast, we can see that the multi-objective planning method achieves not only excellent tracking performance but also coverage of the search area. Notably, this feat is achieved with only *three agents* and four groups of objects with a total of 20 fast moving mobile objects.

Agent trajectories reveal that agent rewarding tracking performance alone ($V_1$) leads to teams moving towards and tracking the immediately visible objects. For example, we can observe in Figure 5.9 that all three agents almost always follow the immediately visible object groups when planning decision are made using only the $V_1$ value

function aiming to achieve improved tracking performance. Further, as two groups of objects take the agents away from the other opposing groups, the opposing groups are rarely ever tracked. In contrast, in Figure 5.9, we can clearly see the ***three agents*** using the multi-objective value function for planning, traversing between the ***four*** groups of 20 fast moving objects to track all of the targets.

### 5.5.7 Overall performance with a vision-based sensor

We presented the results from our four scenarios using a range and bearing sensor in Section 5.5 (see Table 5.1). Here, Table 5.2 compares results for scenario 1, 2, 3 and 4 collected from 20 MC runs where the agent use an onboard *vision* sensor with detection range $r_d = 200$ m.

Our observations for the vision based sensor is similar to the *range and bearing* based sensor experiments. It is expected that the average search area entropy is smallest for $V_2$ since it encourages agents to explore the search area. Consequently, $V_2$ can also be seen to achieve better performance in terms of OSPA cardinality—**OSPA Card** compared to planning with value function $V_1$ (for tracking). In contrast, we can see

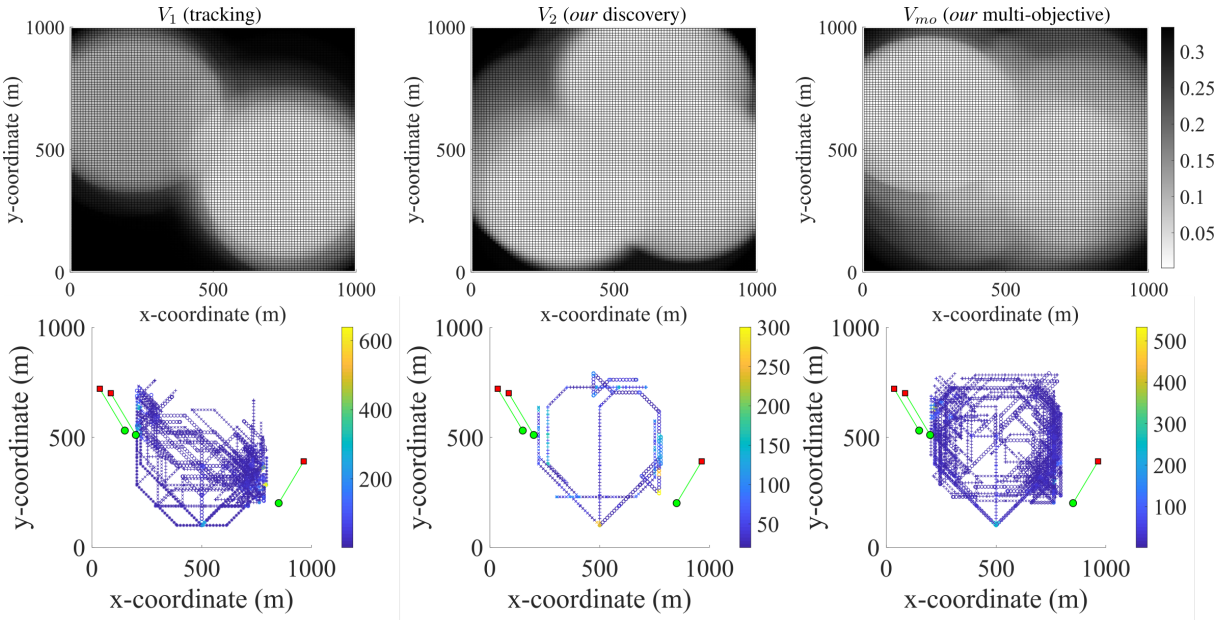


**Figure 5.7. Scenario 1 (FastMoving) using vision-based sensors.** 2D grid occupancy probability (top) and heat map (bottom) of the trajectories of **three agents** to search and track **three** fast moving objects over 20 MC runs using *range and bearing* based sensor with detection range $r_d = 200$ m.

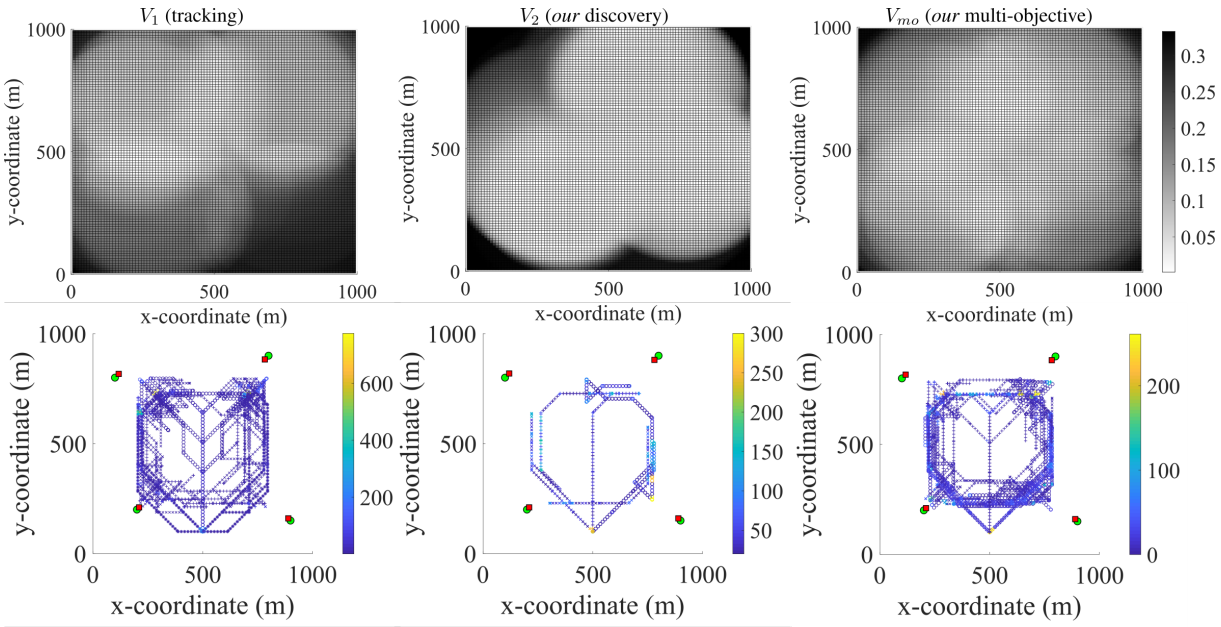


**Figure 5.8. Scenario 2 (LateBirth) using vision-based sensors.** 2D grid occupancy probability (top) and heat map (bottom) of the trajectories of **three agents** to search and track four slow moving objects over 20 MC runs using *range and bearing* based sensor with detection range $r_d = 200$ m.

that multi-agent planning decisions that value function only tracking accuracy, $V_1$, achieves improved results for object localisation accuracy only (low **OSPA Loc** results) but at the expense of missing objects often out of the range of the sensors (as seen by significantly large **OSPA Card** results). Most importantly, our results verifies that multi-agent planning with the multi-objective value function $V_{mo}$ performs the best in 6 out of 8 cases in terms of *overall* tracking and cardinality accuracy (reported by **OSPA Dist**). We can expect this result in our challenging scenarios since $V_{mo}$ not only value functions agents for undertaking the discovery of new objects but also value functions agents for accurately tracking discovered objects.

We observe that for Scenario 2 (**LateBirth**), when the number of agents is small ($S = 3$), $V_2$ performs better than $V_{mo}$. This is reasonable since our discovery value function, $V_2$, encourages planing decisions that disperses agents to regions with high entropy or less information. Consequently, the late birth objects are instantly detect. In contrast, planning with $V_{mo}$ value function needs the agent team to complete two tasks (*i.e.*, tracking and discovering) simultaneously and with limited number of agents ($S = 3$). Now, the discovery of the two groups of late birth objects occurs very late. This is evident in the **OSPA Loc** result being better for $V_{mo}$ compare to $V_2$ (10.1 vs 13.6) while

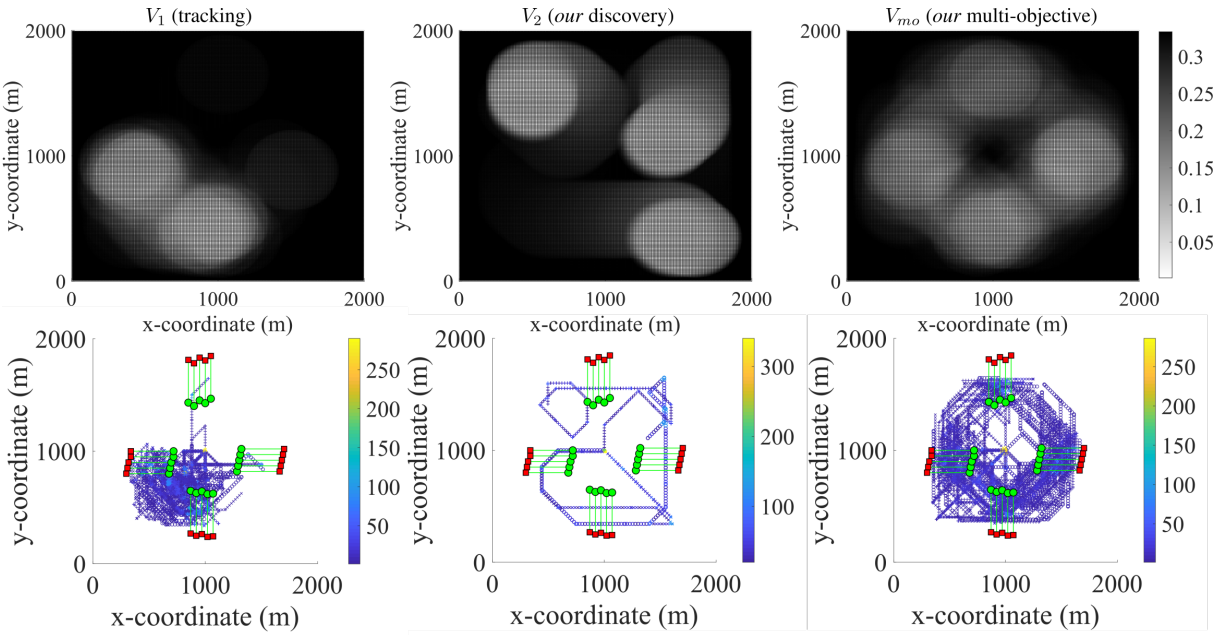


**Figure 5.9. Scenario 4 (Explosion) using vision-based sensors.** 2D grid occupancy probability (top) and heat map (bottom) of the trajectories of **three agents** to search and track **twenty** fast moving objects over 20 MC runs using *vision* based sensor with detection range $r_d = 200$ m. Here object groups $C$ and $D$—see Figure 5.1—are born late.

**OSPA Card** results being worse for $V_{mo}$ compared to $V_2$ (36.4 vs. 29.2) for Scenario 2 (**LateBirth**). However, with more agents ($S = 5$) we can observe the team of agents performing better under the $V_{mo}$ value function planning than with $V_2$ value function alone.

In Scenario 3 (**Opposite**), we can observe planning with value function $V_2$ performs marginally better than the agent team with planning using $V_{mo}$ when $S = 5$. This is because the increased number of agents ($S = 5$) achieves faster exploration over the search area (see the occupancy probability map in Figure 5.3). Now, the object group $B$—see Figure 5.1—are discovered much more quickly with the agents planning for searching with discovery value function $V_2$ to reduce their expected entropy of the search area. This is evident in the **OSPA Card** result being better for the agent team planning using our discovery value function $V_2$ as opposed to our multi-objective value function $V_{mo}$ (3.2 vs. 4.0) while the tracking performance for both teams (using $V_2$ and $V_{mo}$) are nearly identical (13.2 vs. 12.3).

**Table 5.2. Comparing multi-agent planning for tracking mobile objects using our multi-objective value function $V_{mo}$ across four scenarios using vision based sensors with detection range $r_d = 200$ m.** We use planning with the single objective value functions $V_1$ and $V_2$ as baselines and the results are averaged over 20 Monte Carlo experiments.

| | | Scenario 1 (FastMoving) | | | | Scenario 2 (LateBirth) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Indicators | Overall Performance | Tracking Performance | Discovery Performance | | Overall Performance | Tracking Performance | Discovery Performance | |
| *Agents* | *Value functions* | **OSPA Dist (m)** | **OSPA Loc (m)** | **OSPA Card (m)** | **Search Area Entropy (nats)** | **OSPA Dist (m)** | **OSPA Loc (m)** | **OSPA Card (m)** | **Search Area Entropy (nats)** |
| $S = 3$ | $V_1$ | 41.3 | 9.7 | 31.6 | 0.25 | 58.5 | 7.3 | 51.2 | 0.21 |
| | $V_2$ | 26.3 | 15.1 | 11.2 | 0.12 | **42.8** | 13.6 | 29.2 | 0.12 |
| | $V_{mo}$ | **23.2** | 12.1 | 11.1 | 0.16 | 46.5 | 10.1 | 36.4 | 0.16 |
| $S = 5$ | $V_1$ | 31.3 | 10.8 | 20.5 | 0.2 | 52.2 | 8.3 | 43.9 | 0.20 |
| | $V_2$ | 24 | 13.6 | 10.4 | 0.09 | 43.3 | 12.9 | 30.4 | 0.09 |
| | $V_{mo}$ | **23.4** | 12.1 | 11.3 | 0.13 | **39.1** | 10.9 | 28.2 | 0.12 |
| | | Scenario 3 (Opposite) | | | | Scenario 4 (Explosion) | | | |
| $S = 3$ | $V_1$ | 51.5 | 7.8 | 43.7 | 0.26 | 55.3 | 9.3 | 46 | 0.32 |
| | $V_2$ | 22.4 | 17.8 | 4.5 | 0.12 | 54.5 | 37.5 | 17.1 | 0.28 |
| | $V_{mo}$ | **19.4** | 13.4 | 6.0 | 0.16 | **43.4** | 20.8 | 22.6 | 0.32 |
| $S = 5$ | $V_1$ | 51.2 | 8.6 | 42.6 | 0.25 | 43.7 | 12.7 | 31.0 | 0.31 |
| | $V_2$ | **16.4** | 13.2 | 3.2 | 0.09 | 33.4 | 25.4 | 8.0 | 0.25 |
| | $V_{mo}$ | 17.3 | 13.3 | 4.0 | 0.12 | **18.8** | 7.7 | 0.29 | 0.29 |

### 5.5.8 Explore the asymptotic behaviour of tracking performance with increasing agent team's maximum sensor coverage

In Figure 5.4, we studied the behaviour of tracking performance with an increasing number of agents for our planning formulation. In this experiment we consider maximum agent team search area coverage capability and tracking performance.

In practice, sensors have a limited range. In our numerical experiments both the vision sensor and the radio receivers will have a limited range, *i.e.*, $r_d = 200$ m. However, in this experiment, we wanted to understand the performance of our multi-objective planning algorithm when objects are nearly all visible to the agents and systematically remove the constraints imposed by practical detectors. We can expect our single objective value function $V_1$ or $V_2$ to perform better under the theoretical conditions of agent sensors with unlimited detection range.

We recognise that increasing the number of agents, albeit with a range limited sensor is akin to increasing the range of a sensor to be extremely large for a small team of agents. Therefore, we define the *Agent Team's Maximum Sensor Coverage (%)* as $100S \times \pi r_d^2 / (\textit{Search Area})$ %, and investigate the overall performance over in increasing the *Agent Team's Maximum Sensor Coverage (%)*.

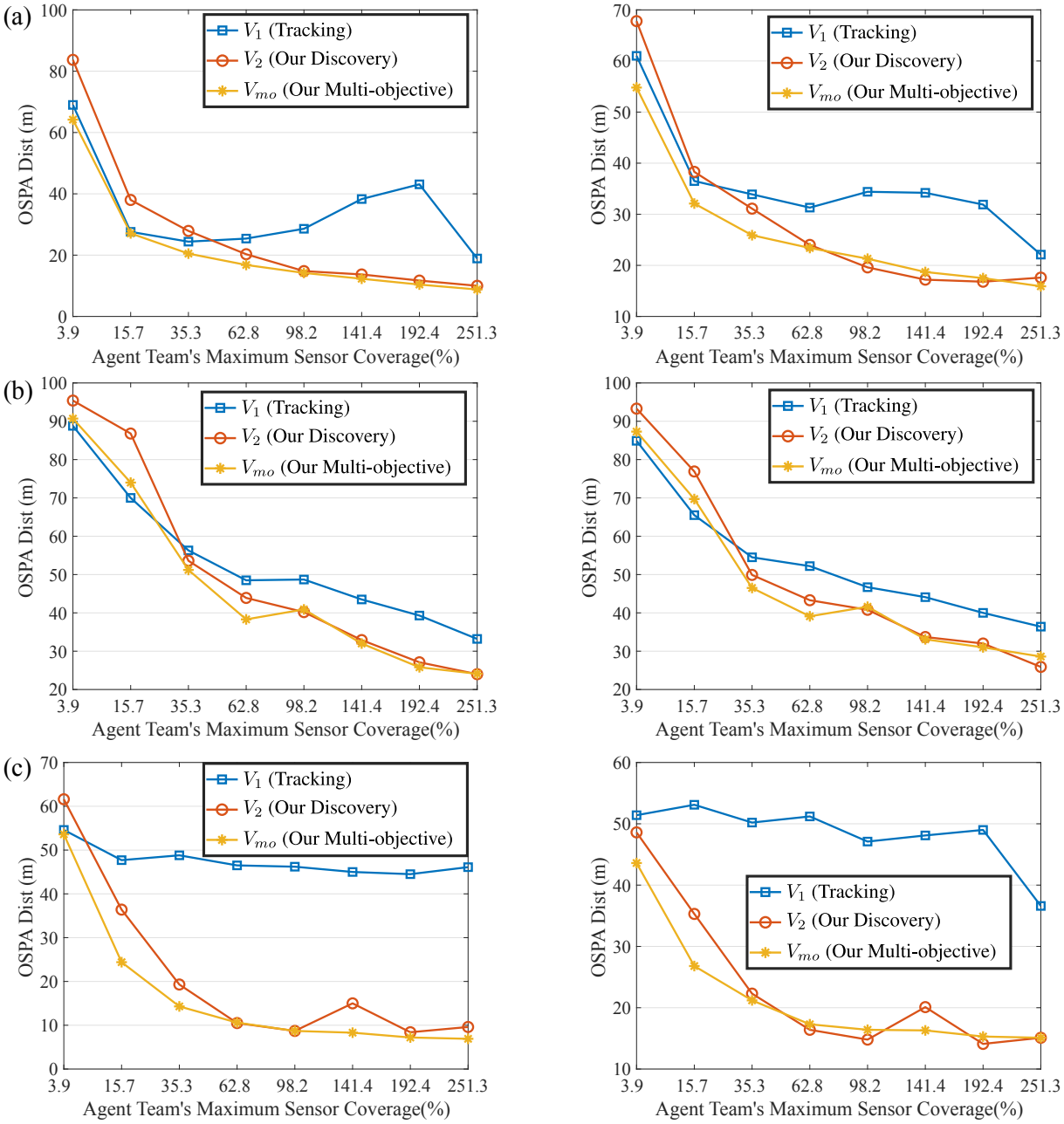


**Figure 5.10. Overall tracking performance over 20 MC runs based on multi-agent planning with our multi-objective value function function $V_{mo}$ compared with the single objective tracking value functions $V_1$ and our discovery $V_2$.** The *Agent Team's Maximum Sensor Coverage (%))* using five agents is increased from 3.9 % to 251.3 % for (a) Scenario 1 (**FastMoving**), (b) Scenario 2 (**LateBirth**), and (c) Scenario 3 (**Opposite**) with (**left**) *range and bearing* based sensors, (**right**) *vision* based sensors.

Figure 5.10 and Figure 5.11 depict the overall performance for planning with our multi-objective value function $V_{mo}$ versus tracking value function $V_1$ and our discovery

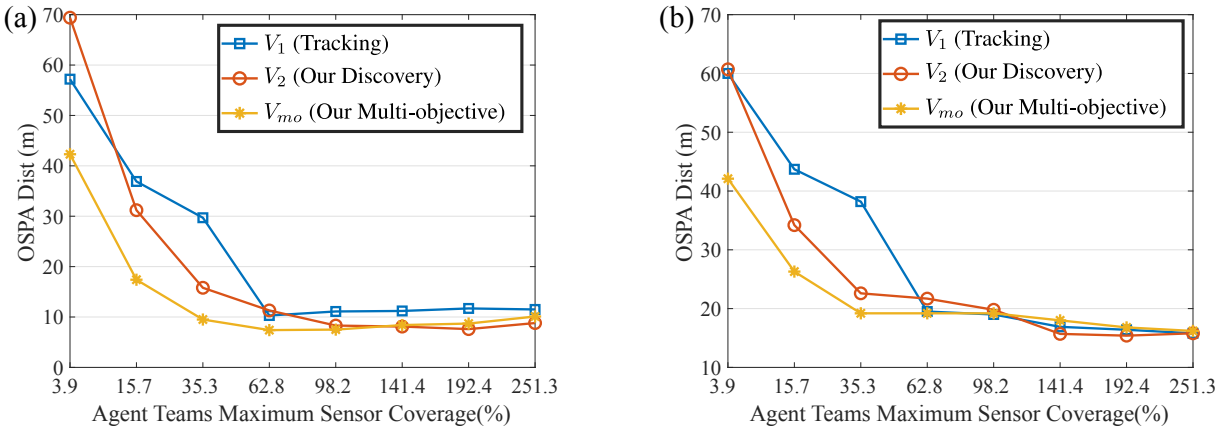


**Figure 5.11. Overall tracking performance over 20 MC runs based on multi-agent planning with our multi-objective value function function $V_{mo}$ compared with the single objective tracking value functions $V_1$ and our discovery value function $V_2$.** The *Agent Team's Maximum Sensor Coverage (%)* using five agents is increased from 3.9 % to 251.3 % for Scenario 4 (**Explosion**) with (a) *range and bearing* based sensors, (b) *vision* based sensors.

value function $V_2$ for the four scenarios. The *Agent Team's Maximum Sensor Coverage (%)* using five agents ($S = 5$) is increased from 3.9 % to 251.3 %; this corresponds to a detection range $r_d$ increase from 50 m to 400 m for scenario 1, 2, and 3 and an $r_d$ increase from 100 m to 800 m for scenario 4.

We can see that planning with value function $V_1$ performs worst and provides inconsistent performance across the four scenarios compared to $V_2$ and $V_{mo}$. Planning with our exploration value function $V_2$ alone generally performs better when the team coverage increases; this is expected. Multi-agent planning with our multi-objective value function $V_{mo}$ almost always performs better or the same as $V_1$ or $V_2$ across all scenarios, especially under the more *practical* situations of low team area coverage ($< 99\%$). Notably, it is difficult for any planning strategy to achieve an overall result better than **OSPA Dist**$\approx$10 m under our experimental settings for the detectors given the realistic process and measurement noise levels we employed.

## 5.6 Conclusion

In this chapter, we have formulated a multi-objective planning approach for multi-agent tracking and searching for mobile objects. We have established that our formulation results in a value function that is monotone and submodular. We

presented a series of extensive experimental results to demonstrate the effectiveness of our method and performance guarantees when using the low-cost greedy algorithm to determine control actions for the multi-agent.

So far, we consider a centralised MPOMDP for multi-agent planning for MOT where scalability can be a limitation. Moreover, the centralised approach can be a serious limitation in real life cases when long distance and a large number of agents (*e.g.*, UAVs) are required to explore a large area. A scalable approach should be a distributed POMDP for MOT, wherein each agent runs its local filter to track multiple objects, and coordinates with other agents to achieve a global objective. However, planning for multiple agents to reach a global goal under a distributed POMDP framework is an NEXP-complete problem (Bernstein et al., 2002). In the next chapter, we focus on solving MOT in a distributed manner where agents are stationary and equipped with limited FoV sensors. Solving this distributed MOT problem is an essential first step towards developing a distributed multi-agent path planning algorithm for MOT.

Chapter 6

# Distributed Multi-object Tracking under Limited FoV Sensors


WE consider the problem of tracking multiple objects using distributed multi-sensors with limited field of views. Specifically, to achieve real-time tracking under limited computing resources, we develop a novel fusion strategy that operates on local multi-object track estimates instead of local multi-object densities. Consequently, the proposed distributed multi-sensor multi-object tracking algorithm is significantly faster than those based on multi-object density fusion. It also achieves better tracking accuracy by directly considering the error of the multi-object track via the Optimal Sub-Pattern Assignment (OSPA) metric. Numerical experiments demonstrate the real-time capability of our proposed solution, in speed and accuracy compared to state-of-the-art fusion rules such as Generalised Covariance Intersection (GCI) in challenging scenarios.

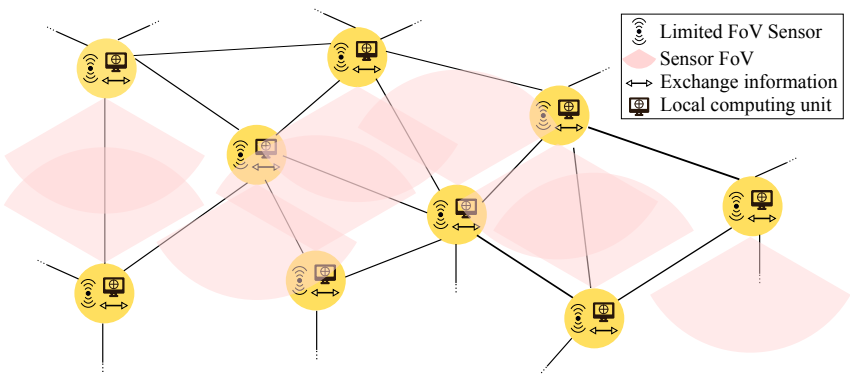


**Figure 6.1.** A distributed sensor network system with limited FoVs.

## 6.1 Motivation and Contribution

Progress in wireless communication technologies and sensing has enabled significant developments in sensing capabilities for sensor networks. These networks are comprising of interconnected nodes with perception, communication and processing capabilities, such as the distributed network illustrated in Fig. 6.1. To fully capitalise on the potential of sensor networks, a properly distributed algorithm should be: i) *scalable* across increasing numbers of nodes; ii) *flexible* to adapt to different scenarios; and iii) *reliable*—avoiding the pitfalls from a single point of failure in a centralised approach (Luo et al., 2006). In the distributed networks, each node can work independently and without a centralised node or knowledge of the network topology.

Distributed MOT (DMOT) is a natural extension of sensor networks. DMOT addresses the practical problems of limited field of view (FoV) or observability of sensing modalities from a single node, such as thermal cameras or radars, to obtain a *comprehensive understanding* of surveillance areas. Such networks are ubiquitous in various smart cyber-physical systems ranging from traffic management, patient care, battlefield surveillance to space exploration (Souza, Nakamura and Pazzi, 2016). The common objective is to combine or *fuse* information from sensors or *multiple nodes* with limited observability to track *multiple objects* under surveillance. In principle, an optimal fusion can be accomplished by preserving marginal and joint distributions from all nodes (Liggins et al., 1997). However, maintaining these quantities demands common information to be shared among all nodes (Mahler, 2000; Battistelli et al., 2013); this hinders the flexibility and scalability of distributed networks. Therefore, a robust (likely sup-optimal) information fusion is essential to counteract the double-counting of information when the common information is

unknown (Üney, Clark and Julier, 2013). In addition, practical considerations on computing resource and communication bandwidth for distributed solutions, means that fusing multi-object distributions (which invariably contain a large number of parameters) should be avoided.

Recently, there has been an emergence of DMOT algorithms developed from the RFS framework. This RFS framework is an emerging paradigm that generalises the classical dynamical systems to set-valued dynamical systems, in which MOT is a multi-object state estimation problem. Consequently, a considerable number of RFS filters have been developed, including *i) non-labelled filters* such as the probability hypothesis density (PHD) filter (Mahler, 2003), cardinalised PHD (CPHD) filter (Mahler, 2007a), multi-Bernoulli (MB) filter (Mahler, 2007b; Vo, Vo and Cantoni, 2009); *ii) labelled filters* such as generalised labelled multi-Bernoulli (GLMB) filter (Vo and Vo, 2013; Vo, Vo and Phung, 2014), and labelled multi-Bernoulli (LMB) filter (Reuter et al., 2014). Under the RFS framework, Generalised Covariance Intersection (GCI[10]) (Mahler, 2000; Hurley, 2002) has emerged as an efficient solution to fuse multi-object densities using the geometric mean from multiple nodes in a distributed network. Subsequently, distributed multi-object tracking filters using GCI under the RFS framework, such as *i) non-labelled GCI-based filters:* PHD-GCI (Üney, Clark and Julier, 2013), CPHD-GCI (Battistelli et al., 2013), MB-GCI (Wang et al., 2016); and *ii) labelled GCI-based filters:* LMB-GCI (Fantacci et al., 2018), Label Free LF-GCI (Li et al., 2018; Yi et al., 2020), Label Matching LM-GCI (Li et al., 2019) have been formulated. In essence, GCI can be considered as an intersection among multi-object densities of multi-sensors, thus, GCI-based filters perform badly when nodes do not *share the same FoV*, as in practice, because GCI is affected by cardinality inconsistency (Üney et al., 2019). Further, GCI fusion in a distributed network setting is computationally inefficient and demands extremely high bandwidth to share multi-object densities among all nodes (Fantacci et al., 2015); these aspects can be prohibitive in practice for real-time tracking with increasing numbers of networked nodes and objects of interest (Gao, Battistelli and Chisci, 2019).

A few remedies for distributed fusion problems with different FoVs have been investigated. For non-labelled GCI-based filters, Li et al. proposed using cluster analysis for PHD-GCI filter (CA-PHD-GCI) (Li et al., 2020). The CA-PHD-GCI filter

[10]GCI is also known as Chernoff fusion (Cover and Thomas, 2012; Chang, Chong and Mori, 2010), Exponential Mixture Density (EMD) (Julier, Bailey and Uhlmann, 2006; Clark et al., 2010; Üney, Clark and Julier, 2013) or Kullback-Leibler Average (KLA) (Battistelli et al., 2013; Battistelli and Chisci, 2014)

solution requires sharing FoV information among nodes and does not generate *object labels (identifications)*, an important feature in MOT problems to obtain object tracks (Vo, Vo and Phung, 2014), since its underlying filter is the non-labelled PHD filter. For labelled GCI-based filters, early efforts to minimising label inconsistency problems (*e.g.*, an object is assigned different labels by different nodes) for sensors *without FoV limitations* have been investigated in (Li et al., 2018, 2019; Yi et al., 2020) for distributed fusion problems. However, the problem of reaching label consensus and minimising label inconsistency problems among multiple nodes under practical settings of limited FoV in distributed fusion remains a challenging problem to solve. Importantly, we observe that GCI-based divergence is not a metric (Van Erven and Harremos, 2014).

In this work, we focus on solving DMOT problems under the practical challenges of sensors with limited FoVs and communication channel bandwidth. Therefore we formulate a computation and bandwidth efficient method to remedy the label inconsistency problem in DMOT when nodes operating with limited FoV sensors do not share the same label space. We propose sharing and fusing local estimates instead of the local multi-object densities in a distributed manner; this fusion strategy allows us to significantly reduce the communications bandwidth among nodes as well as the fusing time (*computational cost*). The track-to-track fusion and association algorithms have been investigated previously in various literature (Chong, Mori and Chang, 1990; Chang, Saha and Bar-Shalom, 1997; Chong et al., 2000; Mori et al., 2002; Mori and Chong, 2003; Kaplan, Bar-Shalom and Blair, 2008; Mori, Chang and Chong, 2014; Tian, Yuan and Bar-Shalom, 2015). However, a common practice of these track-to-track association methods is to assume that there is no false tracks or missed objects (thus the number of local tracks from two nodes are the same) (Mori, Chang and Chong, 2014). Therefore, these previous track-to-track association algorithms are not suitable to handle the complex nature of our considered problem involving a time-varying number of objects with unknown associations and the limited FoV sensors. In particular, we formulate a track consensus algorithm, TC-OSPA$^{(2)}$ based on the similarity measures between estimated tracks. Intuitively, we can understand our approach by considering the fact that the tracks, albeit partial, seen over time by multiple nodes for the same object ought to be alike. This requires considering a track consensus problem with, ideally, guarantees on object track similarity measure using Optimal Sub-Pattern Assignment$^{(2)}$ (Beard, Vo and Vo, 2017, 2018) — a metric — between two estimated tracks. Consequently, we establish a performance bound for the label inconsistency problem by exploring the metric property of OSPA$^{(2)}$ distance.

Further, we develop a label consensus algorithm to minimise the label inconsistency problems since local nodes are equipped with limited FoV sensors, and mismatched labels among nodes are common, especially when objects are moving from one node's FoV to another node's FoV.

**Our contributions::** ***i)*** a *fast* and *efficient* distributed fusion algorithm for multi-sensor with different FoVs is devised; ***ii)*** a *novel label consensus* algorithm is developed to minimise the label inconsistency problems due to limited FoV sensors; ***iii)*** Simulation results confirm the effectiveness of our proposed method that considerably outperforms the GCI-based approaches in terms of accuracy and fusing time.

## 6.2 Background

This section presents a brief background needed in our work, including our notational convention, a background about metrics, and the descriptions of a distributed sensor network.

### 6.2.1 Notation

The notational convention in (Vo, Vo and Phung, 2014) is adopted, lowercase letters (*e.g.*, $x, \mathbf{x}$) denote single-object states, while uppercase letters (*e.g.*, $X, \mathbf{X}$) denote multi-object states. Labelled states and their distributions are represented in bold letters (*e.g.*, $\mathbf{X}$, $\boldsymbol{\pi}$), while spaces are denoted in blackboard letters (*e.g.*, $\mathbb{X}, \mathbb{L}$). A labelled single-object state $\mathbf{x}$ is comprising of its unlabelled kinematic state $x$ and label $l$, *i.e.*, $\mathbf{x} = (x, l)$. Let $\mathcal{L} : \mathbb{X} \times \mathbb{L} \to \mathbb{L}$ be the label projection $\mathcal{L}((x, l)) = l$, thus $\mathcal{L}(\mathbf{X}) = \{\mathcal{L}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}$ is the set of labels of $\mathbf{X}$. Further, for a given set $X$, $\mathcal{F}(X)$ denotes the class of finite subsets of $X$, while $1_X(\cdot)$ denotes the indicator function of $X$, and its cardinality is denoted by $|X|$. Through out the texts, the term metric or distance is used interchangeably with the same meaning.

### 6.2.2 Multi-object tracking metrics

In this subsection, we revisit the metric property, and a few notable metrics used in multi-object tracking.

**Metric property** The metric has a precise mathematical meaning. A function $d : \mathcal{F}(\mathbb{X}) \times \mathcal{F}(\mathbb{X}) \to [0, \infty)$ is a metric if it meets the following conditions:

1) $d(x, y) = 0$ if and only if $x = y$ (identity),

2) $d(x, y) = d(y, x)$ (symmetry),

3) $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality).

**Wasserstein metric** Consider two multi-object states $X = \{x^{(1)}, \ldots, x^{(m)}\} \in \mathcal{F}(\mathbb{X})$ and $Y = \{y^{(1)}, \ldots, y^{(n)}\} \in \mathcal{F}(\mathbb{X})$, and a parameter $p \in [1, \infty)$. Hoffman and Mahler (Hoffman and Mahler, 2004) introduced the Wasserstein metric $d_p^W(\cdot, \cdot)$ for empirical densities, given by:

$$d_p^W(X, Y) = \min_C \Big( \sum_{i=1}^{m} \sum_{j=1}^{n} C_{i,j} d(x_i, y_j)^p \Big)^{1/p}, \tag{6.1}$$

where the minimum is taken over all $m \times n$ transportation matrices $C = (C_{i,j})$. An $m \times n$ matrix $C$ is a transportation matrix if $C_{i,j} \geq 0 \; \forall i = 1, \ldots, m; \; j = 1, \ldots, n$, and

$$\sum_{i=1}^{m} C_{i,j} = \frac{1}{n}, \sum_{j=1}^{n} C_{i,j} = \frac{1}{m}. \tag{6.2}$$

The Earth mover's distance in (Levina and Bickel, 2001) is a special case of the Wasserstein metric when $p = 1$, which is the minimum cost of turning one earth pile to the other. The Wasserstein metric encounters a few important problems (see (Schuhmacher, Vo and Vo, 2008) for more details), *e.g.*,

*i)* *The inconsistency of the metric*: the performance of the Wasserstein metric depends on the equitable distribution of the number of objects between two multi-object states, thus it occasionally fails to detect cardinality errors.

*ii)* *Geometry dependent behavior:* the cardinality error is penalised more heavily when objects are further apart.

**OSPA metric** The OSPA metric is proposed in (Schuhmacher, Vo and Vo, 2008) to alleviate the aforementioned problems of the Wasserstein metric. Let $d_p^{(c)}(\cdot, \cdot)$ be the OSPA metric with order $p \in [1, \infty)$ and cutoff $c \in (0, \infty)$. Consider two multi-object

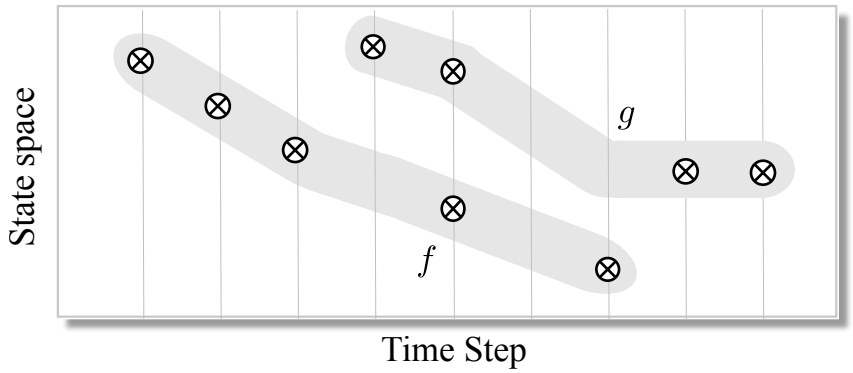


**Figure 6.2.** A sketch for track definition with two fragmented tracks $f$ and $g$.

states $X = \{x^{(1)}, \ldots, x^{(m)}\} \in \mathcal{F}(\mathbb{X})$ and $Y = \{y^{(1)}, \ldots, y^{(n)}\} \in \mathcal{F}(\mathbb{X})$. If $m \leq n$, $d_p^{(c)}(X, Y)$ is defined as

$$d_p^{(c)}(X, Y) = \left( \frac{1}{n} \left( \min_{\pi \in \Pi_n} \sum_{i=1}^{m} \bar{d}^{(c)}(x^i, y^{\pi(i)})^p + c^p(n-m) \right) \right)^{1/p}, \tag{6.3}$$

where $\bar{d}^{(c)}(x, y) = \min(c, d(x, y))$, in which $d(\cdot, \cdot)$ is an arbitrary metric on the single object state space of $\mathbb{X}$. If $m > n$, then $d_p^{(c)}(X, Y) \triangleq d_p^{(c)}(Y, X)$. Since the factor of $1/n$ is used in (6.3), the OSPA metric can be interpreted as the smallest localisation and cardinality per-object error. The cut-off $c$ is a parameter to emphasise on the localisation error if $c$ is small or on the cardinality error if $c$ is large.

**Base distance between two tracks — a metric** Let $\mathbb{T} = \{1, \ldots, K\}$ be a finite space of time indices, from the start time at 1 to the end time at $K$. Let $\mathbb{U} \triangleq \{f : \mathbb{T} \to \mathbb{X}\}$ be a space of all functions from $\mathbb{T}$ to $\mathbb{X}$. An element of $f \in \mathbb{U}$ is defined as a *track*, and its domain $\mathcal{D}_f \subset \mathbb{T}$ is the set of time instants that the object exists. This track definition (as illustrated in Fig. 6.2) encompasses the so-called fragmented tracks (*e.g.*, tracks of estimated objects with holes in their domains due to disappearance/reappearance) and the continuous tracks (so-called trajectories), *e.g.*, tracks of ground truth objects without holes in their domains.

Let $f$ and $g$ be the two tracks of interest on the space $\mathcal{F}(\mathbb{U})$. The base distance between two tracks, $\tilde{d}^{(c)}(\cdot, \cdot)$, is defined as the average OSPA distance between the set of states of tracks $f$ and $g$, over its domain $t \in \mathcal{D}_f \cup \mathcal{D}_g$, *i.e.*,

$$\tilde{d}^{(c)}(f, g) = \sum_{t \in \mathcal{D}_f \cup \mathcal{D}_g} \frac{d^{(c)}(\{f(t)\}, \{g(t)\})}{|\mathcal{D}_f \cup \mathcal{D}_g|} \tag{6.4}$$

if $\mathcal{D}_f \cup \mathcal{D}_g \neq \emptyset$, and $\tilde{d}^{(c)}(f,g) = 0$, if $\mathcal{D}_f \cup \mathcal{D}_g = \emptyset$. Here, $\{f(t)\}$ is a singleton if $t \in \mathcal{D}_f$, and empty otherwise (likewise for $\{g(t)\}$). Thus, the parameter $p$ in (6.3) is removed due to redundancy. As shown in (Beard, Vo and Vo, 2020), the base distance between two tracks is a metric with all aforementioned metric properties.

**OSPA$^{(2)}$ metric for tracks** The base distance defined in (6.4) is a metric, and confined by the cut-off value $c$. Thus, it can be served as a base distance of the original OSPA metric in (6.3). The resulting metric is called OSPA-on-OSPA or OSPA$^{(2)}$ metric. The OSPA$^{(2)}$ metric can be interpreted as the time-average per-track error, which considers errors in localisation, cardinality, as well as track fragmentation and labelling errors (*i.e.*, label inconsistency problems) (Beard, Vo and Vo, 2020).

### 6.2.3 Distributed sensor network description

Consider a distributed heterogeneous network in Fig. 6.1 described by an undirected graph $\mathcal{G} = (\mathcal{S}, \mathcal{A})$ where $\mathcal{S}$ is the set of nodes and $\mathcal{A} \in \mathcal{S} \times \mathcal{S}$ is the set of arcs representing connections among nodes. Particularly, $(a,b) \in \mathcal{A}$ if node $a$ can receive information from node $b$. Then, for each node $a$, let $\mathcal{S}^{(a)} \triangleq \{\forall b \in \mathcal{S} : b \neq a \text{ and } (a,b) \in \mathcal{A}\}$ be the set of neighbours that node $a$ can receive data from.

The network of nodes $\mathcal{S}$ is conducting surveillance over a large area to detect and track an unknown and time-varying number of mobile objects. We assume that each node is equipped with a local computational unit capable of, for example, computing its local probability density and its resulting local estimate, as well as a transceiver for communicating reliably with their neighbours through a range and bandwidth limited communication channel. In this context, a typical ad-hoc network exists between the nodes where a repeated message passing mechanism can propagate the local information of each node across the network. A node is equipped with a limited field-of-view (FoV) sensor subjected to false-alarms and misdetections, as in practice. Further, the associations between objects and measurements are unknown and must be estimated online.

The network of interest has no central fusion node and its nodes operate without knowledge of the network topology. The objective in this problem is to reach an agreement amongst the local estimates of the set of nodes $\mathcal{S}$; this is a fundamental problem for a network. Our interest is in networked estimation algorithms scalable

with respect to network size and permitting each node to operate without knowledge of the dependence between its own information and the information from other nodes.

## 6.3 Information Fusion using Track Consensus

To this end, we consider that each local node computes its local LMB filter and its resulting local estimate. The goal is to reach the estimate consensus among nodes, comprising of: *i) spatial consensus* (*i.e.*, object's positions) and *ii) label consensus* (*i.e.*, object's identifications) of local estimates. Note that the notion of consensus requires a physically meaningful distance (*i.e.*, OSPA$^{(2)}$) with metric properties to measure the similarity among local estimates.

In this section, we present our proposed distributed information fusion method for local estimates from the local LMB filters of a network of $|\mathcal{S}|$ distributed nodes. The fundamental idea is to find matched tracks by measuring the similarity between two tracks over a time window (*e.g.*, 10-scan), instead of relying on the state of a single scan—measurement time step—as in previous work Li et al. (2019). Additionally, since the sensor on each node has a limited FoV, the fusion algorithm should only fuse the matched tracks to enhance tracking accuracy while preserving the unmatched ones to enlarge detectability. Further, the fusion algorithm needs to address the challenging problem of label inconsistency resulting from the limited FoV sensors in the presence of fragmented tracks to achieve label consensus.

First, for the sake of simplicity, we consider information fusion for the case of two nodes. We present how to use the base distance of OSPA$^{(2)}$ metric as a distance metric of two tracks. Second, we derive a performance bound for the label inconsistency problem. Third, we present the spatial consensus method. Forth, we propose a label consensus algorithm to counteract the label inconsistency problem. Last, a message-passing mechanism coupling with a pair-wise fusing algorithm is proposed when the number of nodes $|\mathcal{S}| > 2$.

### 6.3.1 Track matching for two nodes

The previous work in (Li et al., 2019) using LM-GCI only considers label matching at one instance in time. Further, LM-GCI can only perform well in full FoV scenarios when all of the nodes can observe all of objects. For limited FoV cases, there are

inconsistent labels in case the objects only appear in one node's FoV but not the others, resulting that the mismatched objects are not detected. In this work, we propose using the tracks of detected objects to match among nodes which considers the histories of objects for a better match. Further, the unmatched objects are maintained for the limited FoV scenarios.

Considering two nodes $a, b \in \mathcal{S}$ with its local estimated tracks over a time interval $[j,k]$ are $\mathbf{X}_{j:k}^{(a)}$ and $\mathbf{X}_{j:k}^{(b)}$, respectively. At time $k$, the aim is to compute a reported estimate $\mathbf{X}_k^{\text{rep}}$ between two nodes. Because the data fusion is conducted for existing local estimates at time $k$, we consider track matching for the existing tracks at time $k$ only. Let $L_k^{(a)}$ and $L_k^{(b)}$ be the sets of labels at time $k$ of the two respective local estimates, $a$ and $b$. Let $l \in L_k^{(a)}$ and $l' \in L_k^{(b)}$ be two labels of $a$ and $b$ local estimates, respectively. The corresponding domains are $\mathcal{D}^{(l)} \subseteq [j,k]$ and $\mathcal{D}^{(l')} \subseteq [j,k]$, with the respective tracks are $\mathbf{x}^{(l)}$ and $\mathbf{x}^{(l')}$.

Without loss of generality, assume $|L_k^{(a)}| \leq |L_k^{(b)}|$. Let $\tau : L_k^{(a)} \to L_k^{(b)}$ be a track matching map, which is an injective function mapping unique elements of its domain to unique elements of its codomain such that $\tau(l_1) = \tau(l_2)$ if and only if $l_1 = l_2$. The set $\mathbb{T}$ of all such track matching maps is called the track matching map space. The objective is to find the optimal matching that minimises summation of the OSPA$^{(2)}$ base distances between estimated tracks from two local estimates:

$$\tau^* = \underset{\tau \in \mathbb{T}}{\operatorname{argmin}} \sum_{\forall l \in L_k^{(a)}} \tilde{d}^{(c)}(\mathbf{x}^{(l)}, \mathbf{x}^{(\tau(l))}). \tag{6.5}$$

This can be achieved by solving a ranked assignment problem (Vo, Vo and Phung, 2014). The cost matrix of an optimal matching problem is the $|L_k^{(a)}| \times |L_k^{(b)}|$ matrix:

$$C = \begin{bmatrix} C_{1,1} & \cdots & C_{1,|L_k^{(b)}|} \\ \vdots & \ddots & \vdots \\ C_{|L_k^{(a)}|,1} & \cdots & C_{|L_k^{(a)}|,|L_k^{(b)}|} \end{bmatrix}, \tag{6.6}$$

where for $m \in \{1, \ldots, |L_k^{(a)}|\}$ and $n \in \{1, \ldots, |L_k^{(b)}|\}$,

$$C_{m,n} = \tilde{d}^{(c)}(\mathbf{x}^{(l_m)}, \mathbf{x}^{(l'_n)}). \tag{6.7}$$

In particular, let $H$ be an assignment matrix comprised of 0 or 1 elements such that the sum of every row and every column is either 0 or 1. Here, $H_{m,n} = 1$ if $l_m$ is matched to

$l'_n$, and $H_{m,n} = 0$ if otherwise. The assignment matrix cost $H$ is the joint costs of every label in $L_k^{(b)}$ to every label in $L_k^{(a)}$, which can be written as the Frobenius inner product, such that

$$\text{tr}(H^T C) = \sum_{m=1}^{|L_k^{(a)}|} \sum_{n=1}^{|L_k^{(b)}|} C_{m,n} H_{m,n}.$$

The ranked assignment problem can be determined using the Hungarian algorithm in strongly polynomial time (Kuhn, 1955; Munkres, 1957). In the multi-object tracking context, as in this chapter, a Murty's algorithm (Murty, 1968) with $\mathcal{O}(|L_k^{(b)}|^4)$ complexity is implemented. Beside that, other efficient assignment algorithms (Miller, Stone and Cox, 1997; Pedersen, Nielsen and Andersen, 2008) with $\mathcal{O}(|L_k^{(b)}|^3)$ could be considered.

**Remark 2.** *The computational time of the ranked assignment using TC-OSPA$^{(2)}$ is considerably smaller than LM-GCI proposed in (Li et al., 2019). The reason is that for LMB filter, the local estimate is extracted from the local LMB density for any labels with existence probabilities higher than a predefined threshold (typically 0.5). Thus, the label space of the local estimate used in TC-OSPA$^{(2)}$ is substantially smaller than the label space of the LMB density used in LM-GCI.*

### 6.3.2 Performance bound for label inconsistency

In this subsection, we investigate the condition to ensure there is no label inconsistency when performing the optimal assignment algorithm in (6.5). We have the following propositions (see the appendix for proofs).

**Proposition 6.1.** *Given two objects with its ground truth trajectories $\mathbf{y}_{j:k}^{(1)}, \mathbf{y}_{j:k}^{(2)}$. Let $\mathbf{x}^{(l_1)}, \mathbf{x}^{(l_2)}$ and $\mathbf{x}^{(l'_1)}, \mathbf{x}^{(l'_2)}$ be their estimated tracks from node a and node b, respectively. Let the estimated trajectory error bound using the base distance of OSPA$^{(2)}$ for node a and node b over the time interval $[j,k]$ is $E_{a,j:k}$ and $E_{b,j:k}$, respectively. Let $E_{j:k} = \max(E_{a,j:k}, E_{b,j:k})$, that is for any $m = 1, 2$,*

$$\breve{d}^{(c)}(\mathbf{x}^{(l_m)}, \mathbf{y}_{j:k}^{(m)}) \leq E_{j:k}, \text{ and } \breve{d}^{(c)}(\mathbf{y}_{j:k}^{(m)}, \mathbf{x}^{(l'_m)}) \leq E_{j:k}.$$

*Then the condition for no label inconsistency in (6.5) is that*

$$\breve{d}^{(c)}(\mathbf{y}_{j:k}^{(1)}, \mathbf{y}_{j:k}^{(2)}) > 4E_{j:k}. \tag{6.8}$$

*Proof.* Since the base distance of OSPA$^{(2)}$ is a metric, applying triangle inequality, we have:

$$\begin{aligned}\breve{d}^{(c)}(\mathbf{x}^{(l_m)}, \mathbf{x}^{(l'_m)}) &\leq \breve{d}^{(c)}(\mathbf{x}^{(l_m)}, \mathbf{y}^{(m)}_{j:k}) + \breve{d}^{(c)}(\mathbf{y}^{(m)}_{j:k}, \mathbf{x}^{(l'_m)}), \\ &\leq 2E_{j:k}. \end{aligned} \tag{6.9}$$

Similarly, using triangle inequality, we have

$$\begin{aligned}\breve{d}^{(c)}(\mathbf{x}^{(l_1)}, \mathbf{x}^{(l'_2)}) &\geq \breve{d}^{(c)}(\mathbf{y}^{(1)}_{j:k}, \mathbf{x}^{(l'_2)}) - E_{j:k}, \\ &\geq \breve{d}^{(c)}(\mathbf{y}^{(1)}_{j:k}, \mathbf{y}^{(2)}_{j:k}) - 2E_{j:k}. \end{aligned} \tag{6.10}$$

So the condition for no label switching is that:

$$\begin{aligned}&\breve{d}^{(c)}(\mathbf{x}^{(l_1)}, \mathbf{x}^{(l'_2)}) > \breve{d}^{(c)}(\mathbf{x}^{(l_m)}, \mathbf{x}^{(l'_m)}), \\ \Leftrightarrow\ &\breve{d}^{(c)}(\mathbf{y}^{(1)}_{j:k}, \mathbf{y}^{(2)}_{j:k}) > 4E_{j:k}. \end{aligned} \tag{6.11}$$

□

The following definition is needed to explore $E_{j:k}$.

**Definition 1** (The empirical existence probability)**.** A single object with state $y \in \mathbb{X}$ can be estimated using any filter after applying measurements by either a single object with state $x \in \mathbb{X}$ — an object detection, occurring with probability $p_X(y) \in [0, 1]$ or a missed detection, no estimated object at all, occurring with probability $1 - p_X(y)$. Here, $p_X(\cdot)$ is called an empirical detection probability[11]. Over the time period $[j : k]$, the empirical existence probability of an object $\mathbf{y}^{(m)}_{j:k}$ from node $a$ with its estimated track $\mathbf{x}^{(l_m)}$ is

$$p^{(a)}_{X,j:k}(\mathbf{y}^{(m)}_{j:k}) = \frac{|\mathcal{D}^{(l_m)}|}{n}. \tag{6.12}$$

where $\mathcal{D}^{(l_m)}$ is the domain of track $\mathbf{x}^{(l_m)}$, and $n = k - j + 1$.

**Proposition 6.2.** *Let $p^{(a)}_{X,j:k}(\cdot)$ and $p^{(b)}_{X,j:k}(\cdot)$ be the minimum empirical existence probability of node a and node b for all objects, in the period $[j, k]$, and $p^{\min}_{X,j:k} = \min(p^{(a)}_{X,j:k}(\cdot), p^{(b)}_{X,j:k}(\cdot))$.*

[11] The empirical existence probability $p_X$ is different from the detection probability $p_D$ in the sense that $p_D$ is related to detectability of a sensor measurement while $p_X$ is related to the detectability based on the filter's posterior estimate. For any good filters, we can expect $0 \leq p_D \leq p_X \leq 1\ \forall p_D$, and $p_X$ is monotonically increasing to $p_D$.

*Let the state estimation error bound for node $a$ and node $b$ over period $[j,k]$ be $\varepsilon_{a,j:k}, \varepsilon_{b,j:k}$ respectively, and let $\varepsilon_{j:k}^{(c)} = \min(c, \max(\varepsilon_{a,j:k}, \varepsilon_{b,j:k})) \leq c$, that is for any $m = 1, 2$,*

$$d^{(c)}(\{\mathbf{x}_i^{(l_m)}\}, \{\mathbf{y}_i^{(m)}\}) = \begin{cases} c & i \notin \mathcal{D}^{(l_m)} \\ \min(d(\mathbf{x}_i^{(l_m)}, \mathbf{y}_i^{(m)}), c) \leq \varepsilon_{j:k}^{(c)} & i \in \mathcal{D}^{(l_m)} \end{cases},$$

$$d^{(c)}(\{\mathbf{x}_i^{(l'_m)}\}, \{\mathbf{y}_i^{(m)}\}) = \begin{cases} c & i \notin \mathcal{D}^{(l'_m)} \\ \min(d(\mathbf{x}_i^{(l'_m)}, \mathbf{y}_i^{(m)}), c) \leq \varepsilon_{j:k}^{(c)} & i \in \mathcal{D}^{(l'_m)} \end{cases}$$

*Then we have:*

$$E_{j:k} = \varepsilon_{j:k}^{(c)} p_{X,j:k}^{\min} + c(1 - p_{X,j:k}^{\min}). \tag{6.13}$$

*Proof.* Based on the Definition 1, we have:

$$p_{X,j:k}^{(a)}(\mathbf{y}_{j:k}^{(m)}) = \frac{|\mathcal{D}^{(l_m)}|}{n} = p_{X,j:k}^{\min} + \epsilon, \tag{6.14}$$

where $\epsilon \geq 0$.

According to the definition of the base distance of OSPA$^{(2)}$:

$$\begin{aligned} \tilde{d}^{(c)}(\mathbf{x}^{(l_m)}, \mathbf{y}_{j:k}^{(m)}) &= \sum_{i=j}^{k} \frac{d^{(c)}(\{\mathbf{x}_i^{(l_m)}\}, \{\mathbf{y}_i^{(m)}\})}{n} \\ &= \frac{1}{n} \sum_{i \in \mathcal{D}^{(l_m)}} d^{(c)}(\{\mathbf{x}_i^{(l_m)}\}, \{\mathbf{y}_i^{(m)}\}) + \frac{n - |\mathcal{D}^{(l_m)}|}{n} c \\ &\leq p_{X,j:k}^{(a)}(\mathbf{y}_{j:k}^{(m)}) \varepsilon_{j:k}^{(c)} + (1 - p_{X,j:k}^{(a)}(\mathbf{y}_{j:k}^{(m)}))c \\ &\leq (p_{X,j:k}^{\min} + \epsilon)\varepsilon_{j:k}^{(c)} + (1 - p_{X,j:k}^{\min} - \epsilon)c \\ &\leq \varepsilon_{j:k}^{(c)} p_{X,j:k}^{\min} + (1 - p_{X,j:k}^{\min})c - \epsilon(c - \varepsilon_{j:k}^{(c)}) \\ &\leq \varepsilon_{j:k}^{(c)} p_{X,j:k}^{\min} + (1 - p_{X,j:k}^{\min})c. \end{aligned} \tag{6.15}$$

Similarly we have

$$\tilde{d}^{(c)}(\mathbf{x}^{(l'_m)}, \mathbf{y}_{j:k}^{(m)}) \leq \varepsilon_{j:k}^{(c)} p_{X,j:k}^{\min} + (1 - p_{X,j:k}^{\min})c.$$

□

**Remark 3.** *If the time interval $[j,k]$ is long enough, and a good filter is used, we can have $p_{X,j:k}^{\min} \approx p_X^{\min} \geq p_D$. Hence,*

$$E_{j:k} \leq \varepsilon_{j:k}^{(c)} p_D + c(1 - p_D). \tag{6.16}$$

*Further, the estimation error bound* $\varepsilon_{j:k}^{(c)}$ *is often proportional to measurement noise covariance* $R$. *Suppose* $R = \text{diag}(\sigma_x^2, \sigma_y^2)$ *and* $\varepsilon_{j:k}^{(c)} = \min(2\sqrt{\sigma_x^2 + \sigma_y^2}, c)$, *we have:*

$$E_{j:k} \leq \min(2\sqrt{\sigma_x^2 + \sigma_y^2}, c)p_D + c(1 - p_D). \tag{6.17}$$

*Then the condition for no label inconsistency is that*

$$\check{d}^{(c)}(\mathbf{y}_{j:k}^{(1)}, \mathbf{y}_{j:k}^{(2)}) > 4\big[\min(2\sqrt{\sigma_x^2 + \sigma_y^2}, c)p_D + c(1 - p_D)\big]. \tag{6.18}$$

### 6.3.3 Achieving spatial consensus — object's positions

Since each node is equipped with a limited FoV sensor, only tracks that are co-observed by two nodes $a$ and $b$ should be matched. Hence, for all matched pairs $(m, n)$ that $S_{m,n} = 1$, only pairs with associated costs $C_{m,n}$ less than a predefined cost threshold $\mathcal{C}_{\text{match}}$ are considered legitimate. A small $\mathcal{C}_{\text{match}}$ can lead to a smaller number of matched pairs which results in a higher number of unmatched tracks (possibly false tracks) and vice versa. Since the maximum value of $C_{m,n}$ is the cut-off $c$, $\mathcal{C}_{\text{match}}$ should not be larger than $c$. In particular, let $Q_k = [Q_k^{(a)}, Q_k^{(b)}]$ be a 2-column matrix contains all legitimately matched pairs indexes $(m, n)$ such that

$$\begin{aligned} m \in Q_k^{(a)} &\subseteq \{1, \ldots, |L_k^{(a)}|\}, \\ n \in Q_k^{(b)} &\subseteq \{1, \ldots, |L_k^{(b)}|\}, \\ \text{subject to } S_{m,n} = 1 &\text{ and } C_{m,n} < \mathcal{C}_{\text{match}}. \end{aligned} \tag{6.19}$$

Hence, $|Q_k^{(a)}| = |Q_k^{(b)}| \leq \min(|L_k^{(a)}|, |L_k^{(b)}|)$. The detailed algorithm for determining matched pairs is given in Algorithm 6.A.1 (see the appendix).

For the unmatched tracks, which are only observed by one node but not the other, some of these tracks are possibly false tracks created from false-alarm measurements and should be pruned. Thus, we propose that only unmatched tracks with lengths higher than a predefined track length $\mathcal{C}_{\text{length}}$ are preserved. Similar to the traditional thresholding method in extracting raw measurements, a small track length $\mathcal{C}_{\text{length}}$ helps to detect new objects faster with a higher number of false-positive tracks and vice versa. As a result, the reported estimate comprises of two components: *i)* The preserved estimates from two nodes; *ii)* The fused estimate which is observed by two nodes. We have the following spatial fusion procedure for *spatial consensus*[12]:

[12]Object's states are presented in bold letters which include labels; however, their labels could be modified later to achieve label consensus

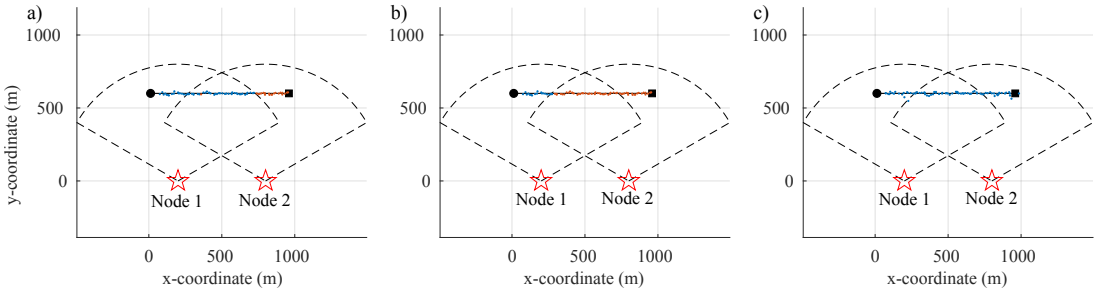


**Figure 6.3. Example 1 — label consensus problem: two nodes with limited FoVs to track a single object.** a) Reported estimate at node 1 before reaching label consensus; b) Reported estimate at node 2 before reaching label consensus; c) Reported estimate at node 1 (likewise for node 2) after reaching label consensus. Here: '○' is location of object's birth; '□' is location of object's death. Colour coding represents labels of the objects.

*1) Pick the preserved estimates:* let $\mathbf{X}_{a,k}^{\text{pres}}$ be the preserved estimate of node $a$ (likewise for node b), which are extracted from the unmatched tracks, subject to a predefined track length, *i.e.*,

$$\begin{aligned} \mathbf{X}_{a,k}^{\text{pres}} \triangleq & \bigcup_{m\in\{1,\ldots,|L_k^{(a)}|\}\setminus Q_k^{(a)}} \mathbf{x}_k^{(l_m)} \\ \text{subject to} \quad & |\mathcal{D}^{(l_m)}| \geq \mathcal{C}_{\text{length}}. \end{aligned} \tag{6.20}$$

*2) Fusion of the matched estimate:* Let $w_a$ and $w_b$ be the fusing weights of two nodes, with $w_a + w_b = 1$ and $w_a, w_b > 0$ (See Section 6.4 for how the weights are selected). For each valid matched pairs $(m,n) \in Q_k$, its corresponding local estimates are $\mathbf{x}_k^{(l_m)} = (x_k^{(l_m)}, l_m)$ and $\mathbf{x}_k^{(l'_n)} = (x_k^{(l'_n)}, l'_n)$. Let $\bar{\mathbf{x}}_k = (\bar{x}_k, \bar{l}) \in \bar{\mathbf{X}}_k$ be the fused estimate, where $\bar{x}_k$ is the fused spatial estimate and $\bar{l}$ is the fused label, *i.e.* (Chong et al., 2000):

$$\bar{x}_k = w_a x_k^{(l_m)} + w_b x_k^{(l'_n)}, \tag{6.21}$$

$$\bar{l} = \begin{cases} l_m, & \text{if the fusion step performs at node } a, \\ l'_n, & \text{if the fusion step performs at node } b. \end{cases} \tag{6.22}$$

*3) Compute the reported estimate:* Let $\mathbf{X}_k^{\text{rep}}$ be the reported estimate at time $k$ comprising of two components: *i)* the preserved estimates of two nodes, *ii)* the fused estimate, *i.e.*,

$$\mathbf{X}_k^{\text{rep}} = \bar{\mathbf{X}}_k \cup \mathbf{X}_{a,k}^{\text{pres}} \cup \mathbf{X}_{b,k}^{\text{pres}}. \tag{6.23}$$

**Remark 4.** *The reported estimate $\mathbf{X}_k^{rep}$ is used for reporting purposes only. For spatial fusion procedures listed above, the local estimate $\mathbf{X}_k$ is maintained independently for the next fusion*

*step, which relies on the local estimate over a time window. The reason is that the proposed spatial fusion procedure is conducted at the local estimate level, thus the local belief density is not updated.*

### 6.3.4 Achieving label consensus — object's identifications

The previous subsection presents our proposed spatial fusion procedure for two nodes with limited FoVs for reaching *spatial consensus*. However, MOT concerns not only object's positions (spatial) but also object's identifications (labels). Therefore, in this subsection, we provide our solution to achieve *label consensus* by minimising the label inconsistency problem in case of limited FoV sensors for two nodes. The reason is that during fusion steps, the preserved estimates from two nodes are included in the final reported estimate (see (6.23)). Hence, care must be taken to ensure mismatched labels between nodes are resolved to achieve the *label consensus*. The following example explains the problem.

**Example 1.** Given a distributed sensor network of two nodes monitoring an area of $[-500, 1500]$ m $\times [0, 1000]$ m. A sensor mounted on each node can only detect objects within its relative angle $[-60°, 60°]$ with detection probability $p_D = 0.98$ and detection range $r_D = 800$ m. The locations of two nodes are $[0, 400]^T$ and $[0, 800]^T$. Each detected object with a kinematic state $x = [p_x, \dot{p}_x, p_y, \dot{p}_y]^T$ results in an observation $z$ of noisy $xy$ positions, given by: $z = [p_x, p_y]^T + v$. Here, $v \sim \mathcal{N}(0, R)$ with $R = \text{diag}(\sigma_x^2, \sigma_y^2)$ where $\sigma_x = \sigma_y = 10$ m. There is one object moving in constant velocity model, given by $x_k = F^{CV} x_{k-1} + q_{k-1}^{CV}$. Here, $F^{CV} = [1, T_0; 0, T_0] \otimes I_2$, $T_0$ is the sampling interval ($T_0 = 1$ s for our experiments), $\otimes$ denotes for the Kronecker tensor product; $I_2$ is the $2 \times 2$ identity matrix; $q_{k-1}^{CV} \sim \mathcal{N}(0, Q^{CV})$ is a $4 \times 1$ zero mean Gaussian process noise, with co-variance $Q^{CV} = \sigma_{CV}^2 [T_0^3/3, T_0^2/2; T_0^2/2, T_0] \otimes I_2$ where $\sigma_{CV} = 5$ m/s$^2$. Each node runs an LMB filter locally. The considered example is provided in Fig. 6.3. After performing the spatial consensus procedure, we have the two following label inconsistency problems:

*1) Label inconsistency of preserved estimates:* Fig. 6.3a depicts the reported estimate at node 1. Although the spatial fusion procedure successfully helps node 1 track the object, even when the object moves out of node 1's FoV, it causes a track fragmentation problem because of assigning a new label for the same object when the object moves out of the FoV of node 1. The reason is that, when the object moves out of node 1's FoV,

it is not detected by node 1 anymore. Thus, we rely on node 2 information to track that object, including node 2's labels. Since node 2 has a different label (depicted in the red colour) for the object that was observed before by node 1 (depicted in the blue colour), the reported estimate yields two different labels for the same object.

*2) Label inconsistency of the matched estimate:* Fig. 6.3b depicts the reported estimate at node 2. Initially, the object is not detected by node 2, hence, it relies on the information from node 1 to track the object, including node 1's label (depicted in the blue colour). When the object moves into the node 2's FoV, node 2 assigns it a new label. During the fusion for the matched estimate (see (6.22)), the fused label is the new label (depicted in the blue colour) from node 2. As a result, the reported estimate has different labels for the same object.

As shown in Example 1, the label inconsistency problems are due to limited FoV sensors. In the following, we present our solution to minimise the label inconsistency problems to reach the label consensus. The main idea is that we first construct an association history between label pairs which are matched. After that, the label consensus can be achieved based on the number of times each label is associated with each other, and which label is born first. In the following discussions, without loss of generality, we consider the fusion procedure performed at node $a$ receiving the local estimate sent from node $b$.

**Update the association history** Let $\Xi_{1:k}^{(a,b)}$ be the association history up to time $k$ between the label set $L_{1:k}^{(a)}$ of node $a$ and $L_{1:k}^{(b)}$ of node $b$. $\Xi_{1:k}^{(a,b)}$ is an $|L_{1:k}^{(a)}| \times |L_{1:k}^{(b)}|$ matrix. For $m \in Q_k^{(a)}$ and $n \in Q_k^{(b)}$ via (6.19), let $i^{(m)} \in \{1, \ldots, |L_{1:k}^{(a)}|\}$ and $i^{(n)} \in \{1, \ldots, |L_{1:k}^{(b)}|\}$ be its corresponding index in label set $L_{1:k}^{(a)}$ of node $a$ and $L_{1:k}^{(b)}$ of node $b$. We update the association history between labels based on how many times each of the labels is matched with each other, *i.e.*,

$$\Xi_{1:k}^{(a,b)}(i^{(m)}, i^{(n)}) := \Xi_{1:k}^{(a,b)}(i^{(m)}, i^{(n)}) + 1. \tag{6.24}$$

See Algorithm 6.A.2 in the appendix for more details.

**Update preserved labels** As illustrated in Fig. 6.3a, the preserved labels sent from node 2 to node 1 (where the fusion procedure is conducted) may need to be relabelled to node 1's own labels to reach the label consensus between two nodes. Hence, we

propose using the association history $\Xi_{1:k}^{(a,b)}$ to update the preserved labels sent from node $b$. In particular, let $L_k^{\text{pres}} = \mathcal{L}(\mathbf{X}_{a,k}^{\text{pres}} \cup \mathbf{X}_{b,k}^{\text{pres}})$ be the preserved labels, $\bar{L}_k = \mathcal{L}(\bar{\mathbf{X}}_k)$ be the matched labels, $i \in \{1, \ldots, |L_k^{\text{pres}}|\}$ be any label index of preserved labels such that $L_k^{\text{pres}}(:, i) \in L_{1:k}^{(b)}$, $n \in \{1, \ldots, |L_{1:k}^{(b)}|\}$ be the corresponding index in label set $L_{1:k}^{(b)}$ for $i$, and $m \in \{1, \ldots, |L_{1:k}^{(a)}|\}$ be the highest value index in label set $L_{1:k}^{(a)}$ of $\Xi_{1:k}^{(a,b)}(:, n)$, resulting corresponding label $l = L_{1:k}^{(a)}(:, m)$. To ensure the label's uniqueness, only label $l \notin \left[L_k^{\text{pres}} \cup \bar{L}_k\right]$ is updated , *i.e.*

$$L_k^{\text{pres}}(:, i) = l \qquad \text{subject to } l \notin \left[L_k^{\text{pres}} \cup \bar{L}_k\right], \tag{6.25}$$

as given in Algorithm 6.A.3 (see the appendix).

**Update reported labels** As shown in Fig. 6.3b, the reported labels may need to be relabelled due to label inconsistency problems for the matched estimate, which are currently based on the matched labels of the local node where the fusion procedure is performed, *i.e.*, node 2 in this example (see (6.22)). However, if the object was born before it entered node 2's FoV, we should use the sent label from node 1 instead of node 2 to reach the label consensus. In particular, let $L_k^{\text{rep}} = \mathcal{L}(\mathbf{X}_k^{\text{rep}})$ be the reported label, $i \in \{1, \ldots, |L_k^{\text{rep}}|\}$ be any label index of reported label set $L_k^{\text{rep}}$ such that $L_k^{\text{rep}}(:, i) \in L_{1:k}^{(a)}$, $m \in L_{1:k}^{(a)}$ be the corresponding index in the label set $L_{1:k}^{(a)}$ from node $a$, and $n \in L_{1:k}^{(b)}$ be the highest value index in the label set $L_{1:k}^{(b)}$ from node $b$ of $\Xi_{1:k}^{(a,b)}(m, :)$, resulting corresponding label $l' = L_{1:k}^{(b)}(:, n)$. Then the reported label is updated based on label's uniqueness constraint and label's time of birth, *i.e.*,

$$L_k^{\text{rep}}(:, i) := l' \text{ subject to } l' \notin L_k^{\text{rep}} \text{ and } l'(1) < L_k^{\text{rep}}(1, i). \tag{6.26}$$

The proposed algorithm is presented in Algorithm 6.A.4 in the appendix, which updates labels from node $a$ to node $b$ under the label's uniqueness constraint and label's time of birth.

**Label consensus data fusion** Fig. 6.3c depicts the reported estimate at node 1 after the label consensus is achieved, *e.g.*, there is no label inconsistency for the single object. The detailed algorithm summarising our proposed fusion method is given in Algorithm 6.1.

**Algorithm 6.1** FuseTwoEstimates

**Input**: $\mathbf{X}_{j:k}^{(a)}$; $\mathbf{X}_{j:k}^{(b)}$; $\Xi_{1:k-1}^{(a,b)}$; $L_{1:k}^{(a)}$; $L_{1:k}^{(b)}$;
**Output**: $\mathbf{X}_k^{\text{rep}}$; $\Xi_{1:k}^{(a,b)}$
$L_k^{(a)} := \mathcal{L}(\mathbf{X}_k^{(a)})$ ; $L_k^{(b)} := \mathcal{L}(\mathbf{X}_k^{(b)})$;
$Q_k :=$ DetermineMatchedPairs$(\mathbf{X}_{j:k}^{(a)}, \mathbf{X}_{j:k}^{(b)}), L_k^{(a)}, L_k^{(b)})$;
$\Xi_{1:k}^{(a,b)} :=$ UpdateAssociationHistory$(\Xi_{1:k-1}^{(a,b)}, L_{1:k}^{(a)}, L_{1:k}^{(b)}, Q_k)$;
Compute $\mathbf{X}_{a,k}^{\text{pres}}$ and $\mathbf{X}_{b,k}^{\text{pres}}$ via (6.20); $\bar{\mathbf{X}}_k$ via (6.21);
$L_k^{\text{pres}} := \mathcal{L}(\mathbf{X}_{a,k}^{\text{pres}} \cup \mathbf{X}_{b,k}^{\text{pres}})$ ; $\bar{L}_k := \mathcal{L}(\bar{\mathbf{X}}_k)$;
$L_k^{\text{pres}} :=$ UpdatePreservedLabels$(\Xi_{1:k}^{(a,b)}; L_{1:k}^{(a)}; L_{1:k}^{(b)}; L_k^{\text{pres}}; \bar{L}_k)$;
$\mathbf{X}_k^{\text{rep}} := \bar{\mathbf{X}}_k \cup \mathbf{X}_{a,k}^{\text{pres}} \cup \mathbf{X}_{b,k}^{\text{pres}}; L_k^{\text{rep}} := \mathcal{L}(\mathbf{X}_k^{\text{rep}})$;
$L_k^{\text{rep}} :=$ UpdateReportedLabels$(\Xi_{1:k}^{(a,b)}; L_{1:k}^{(a)}; L_{1:k}^{(b)}; L_k^{\text{rep}})$;

### 6.3.5 Information fusion for multiple nodes

In the previous subsections, we have presented a new information fusion method for fusing local estimates between two nodes, as provided in Algorithm 6.1. In reality, for a distributed sensor network, the number of nodes $|\mathcal{S}|$ is often larger than two. Thus, it is important to consider data fusion for scenarios when $|\mathcal{S}| > 2$. Although the proposed method can be extended to the case of $|\mathcal{S}| > 2$, the resulting track matching problem is an $|\mathcal{S}|$-dimensional ranked assignment, which is an NP-hard problem. Hence, we relegate it by performing pair-wise matching for two nodes sequentially.

As discussed in Remark 4, the proposed estimates fusion method is performed at a local estimate level extracted from a local density; hence, the local density of each node is not updated. As a result, the current reported estimate does not influence the local estimate in the next scan. Since we use track matching, which relies on the consecutive of track labels over a time period, we need to maintain the local estimate and the reported estimate separately.

In this work, we propose that each node plays as a node in an ad-hoc network such that the node can receive local estimates from all of the other nodes, either directly or indirectly via directly neighbouring nodes as forwarding nodes. Assume the estimates from one node is broadcasted to all other nodes. This is achievable given the significantly low message sizes realised by transmitting local estimates compared to transmitting local densities. In particular, suppose $|L_{\max}|$ be the maximum number of objects seen by the network, *i.e.*, $|L_{\max}| = \max\left([|L^{(1)}|, \ldots, |L^{(\mathcal{S})}|]\right)$, then the order of

magnitude of data that needs to be shared is upper bounded by [13]

$$|\mathcal{S}||L_{\max}|. \tag{6.27}$$

For example, if $|\mathcal{S}| = 20$ nodes, $|L_{\max}| = 100$ objects, and each object state has 4 dimensions (2D environments), and each dimension is represented by an 8-byte floating point value, then the maximum amount of data that needs to be shared by an agent at one time is 64 KB, which is reasonably low to track a large number of objects using 20 distributed nodes. The proposed message-passing mechanism is slightly different from consensus algorithms in the literature (Olfati-Saber, 2007; Stanković, Stanković and Stipanović, 2009; Cattivelli and Sayed, 2010; Calafiore and Abrate, 2009; Üney, Clark and Julier, 2013; Battistelli and Chisci, 2014; Fantacci et al., 2018), wherein local estimates are not fused but only broadcast to all other nodes. The reason is that, consensus algorithms could be too slow to perform in a real-time manner. Further, adopting consensus algorithms may result in label inconsistencies during consensus steps. Although broadcasting local estimates to all nodes requires higher communication cost, it is a good trade-off for achieving a shorter delay and minimising label inconsistency problems due to limited FoV sensors.

Based on the proposed network architecture, the data fusion for multi-sensor is realised in Algorithm 6.2. The Algorithm 6.2 contains two steps: *i)* Each neighbour estimate is fused to the estimate of the node of interest to ensure the label consensus according to the label set of the node of interest; *ii)* Since the label consensus is reached, the fused estimates between neighbour nodes and the nodes of interest are fused together without relabelling by using an empty association history.

## 6.4 Numerical Experiments

In this section, the proposed TC-OSPA$^{(2)}$ fusion method is investigated and compared to other fusion strategies in two scenarios with multi-sensors in distributed network settings. A 2-dimensional search area is adopted for both scenarios to demonstrate the effectiveness of our method. Standard object dynamic and observation models presented in Example 1 are considered. Each object has a survival probability $P_S = 0.98$. Clutter follows a Possion model with an average of 10 clutter per scan. We

[13] The upper bound is only reached in the case that all nodes observe all the objects. In reality, because of limited FoV sensors, this upper bound will not be reached in most cases.

**Algorithm 6.2** FuseMultiEstimates

**Input**: $\{\mathbf{X}_{j:k}^{(a)}\}_{a=1}^{|\mathcal{S}|}$; $\{\Xi_{1:k-1}^{(a,b)}\}_{a,b=1}^{a,b=|\mathcal{S}|}$; $\{L_{1:k}^{(a)}\}_{a=1}^{|\mathcal{S}|}$;
**Output**: $\{\mathbf{X}_{a,k}^{\text{rep}}\}_{a=1}^{|\mathcal{S}|}$; $\{\Xi_{1:k}^{(a,b)}\}_{a,b=1}^{a,b=|\mathcal{S}|}$
**for** $a = 1 : |\mathcal{S}|$ **do**
  $\mathcal{B} := \{1 : |\mathcal{S}|\} \setminus \{a\}$;
  $\mathbf{X}_{\text{temp}} = []$; ▷ temporary reported estimate.
  **for** $i = 1 : |\mathcal{B}|$ **do** ▷ step 1: fusing each neighbour nodes to node $a$.
    $b := \mathcal{B}(i)$;
    $[\mathbf{X}_{\text{temp}}^{(i)}, \Xi_{1:k}^{(a,b)}] :=$
      FuseTwoEstimates$(\mathbf{X}_{j:k}^{(a)}, \mathbf{X}_{j:k}^{(b)}, \Xi_{1:k-1}^{(a,b)}, L_{1:k}^{(a)}, L_{1:k}^{(b)})$;
  **end for**
  $\mathbf{X}_{a,k}^{\text{rep}} := \mathbf{X}_{\text{temp}}^{(1)}$;
  **if** $|\mathcal{B}| > 1$ **then** ▷ step 2: fusing the fused results of neighbour nodes.
    **for** $i = 2 : |\mathcal{B}|$ **do**
      $\mathbf{X}_{a,k}^{\text{rep}} :=$FuseTwoEstimates$(\mathbf{X}_{a,k}^{\text{rep}}, \mathbf{X}_{\text{temp}}^{(i)}, [], [], [])$;
    **end for**
  **end if**
**end for**

use optimal sub-pattern assignment (OSPA) (Schuhmacher, Vo and Vo, 2008) and OSPA$^{(2)}$ (Beard, Vo and Vo, 2017) to measure performance with cut-off $c = 100$ m, order $p = 1$. The OSPA$^{(2)}$ distance at time $k$ is calculated over a 10-scan window ending at $k$ (see (Beard, Vo and Vo, 2017) for more details).

At each local node, an efficient LMB filter with Gaussian Mixtures approach using Gibbs sampling (Vo, Vo and Hoang, 2016) is implemented to track multiple objects. The existence threshold is set at $10^{-3}$, *i.e.*, any state $l$ with existence probability $r^{(l)} < 10^{-3}$ is pruned. Meanwhile, any state $l$ with existence probability $r^{(l)} > 0.5$ is confirmed as an existing object and extracted as an estimate with label $l$. Further, the Adaptive Birth Procedure (ABP) in (Reuter et al., 2014) is implemented. In particular, the birth distribution $\text{ß}_{B,k+1}$ at time $k+1$ is a function of measurement sets $Z_k$, *i.e.*, $\text{ß}_{B,k+1} = \{r_{B,k+1}^{(l)}(z), p_{B,k+1}^{(l)}(x|z)\}_{l=1}^{|Z_k|}$, where

$$r_{B,k+1}^{(l)}(z) = \min\left(r_{B,\max}, \frac{1 - r_{U,k}(z)}{\sum_{\zeta \in Z_k} 1 - r_{U,k}(\zeta)} \lambda_{B,k+1}\right).$$

Here, $r_{U,k}(z)$ is the probability that the measurement $z$ associated to a track in the hypotheses, given by

$$r_{U,k}(z) = \sum_{I_{k-1},\xi,I_k,\theta_k} 1_{\theta_k}(z) w^{(I_{k-1},\xi)} w^{(I_{k-1},\xi,I_k,\theta_k)}, \tag{6.28}$$

where $w^{(I_{k-1},\xi)}w^{(I_{k-1},\xi,I_k,\theta_k)}$ is in (Vo, Vo and Hoang, 2016, eq.14), $\lambda_{B,k+1}$ is the expected number of births at time $k+1$ and $r_{B,\max}$ is the maximum existence probability of a new born object. In the following two scenarios, we set $\lambda_{B,k+1} = 0.5$ and $r_{B,\max} = 0.03$.

For data fusion, since we do not focus on the weight picking problem, the Metropolis weight is implemented, *i.e.*, for each node $a, b \in \mathcal{S}$,

$$w^{(a,b)} = \begin{cases} \frac{1}{1+\max(|\mathcal{S}^{(a)}|,|\mathcal{S}^{(b)}|)}, & a \in \mathcal{S}, b \in \mathcal{S}^{(a)}, \\ 1 - \sum_{b \in \mathcal{S}^{(a)}} w^{(a,b)}, & a \in \mathcal{S}, b = a. \end{cases}$$

The cost threshold for determining matched pairs is set at $\mathcal{C}_{\text{match}} = 100$ m, which is equal to cut-off $c$. Meanwhile, the track length is set $\mathcal{C}_{\text{length}} = 4$, *i.e.*, only tracks with lengths larger or equal to 4 are preserved.

The proposed fusion method (TC-OSPA$^{(2)}$) is compared to CA-PHD-GCI (Li et al., 2020), LM-GCI (Li et al., 2019), and TC-WASS (the track consensus method using the *Wasserstein* metric between two tracks instead of OSPA$^{(2)}$ base distance) in terms of OSPA, OSPA$^{(2)}$ and fusing time. Note that for CA-PHD-GCI fusion strategy, its underlying PHD filter does not report labels for estimated objects, the resulting OSPA$^{(2)}$ cannot be computed. Thus, we propose using a naive label assignment approach for CA-PHD-GCI as the following: *i)* Match the estimated state $X_k$ to the previous estimated state $X_{k-1}$ by OSPA distance using a similar scheme as in Algorithm 6.A.1 except that the track length is equal to 1 and OSPA distance instead of OSPA$^{(2)}$ is evaluated; *ii)* Assign the same labels for the matched estimate of $X_k$; *iii)* Assign newborn labels at time $k$ for unmatched estimates of $X_k$. The report results are calculated and averaged over 100 Monte Carlo (MC) trials.

### 6.4.1 Scenario 1 — Two nodes

In this scenario, we consider a challenging problem of two nodes with limited FoVs to track a time-varying and unknown number of mobile objects in a survey area of $[-500, 1800]$ m $\times$ $[-100, 1000]$ m, with the maximum number of objects is 22. The sensor mounted on each node has limited FoV that can only detection objects within its relative angle of the interval $[-50°, 50°]$ with $p_D = 0.98$ and $r_D = 1000$ m. Two nodes are located at $[300, -100]^T$ and $[1000, -100]^T$. The duration for this scenario is $K = 80$ s with various birth and death objects occurring. The considered scenario settings is illustrated in Fig. 6.4a.

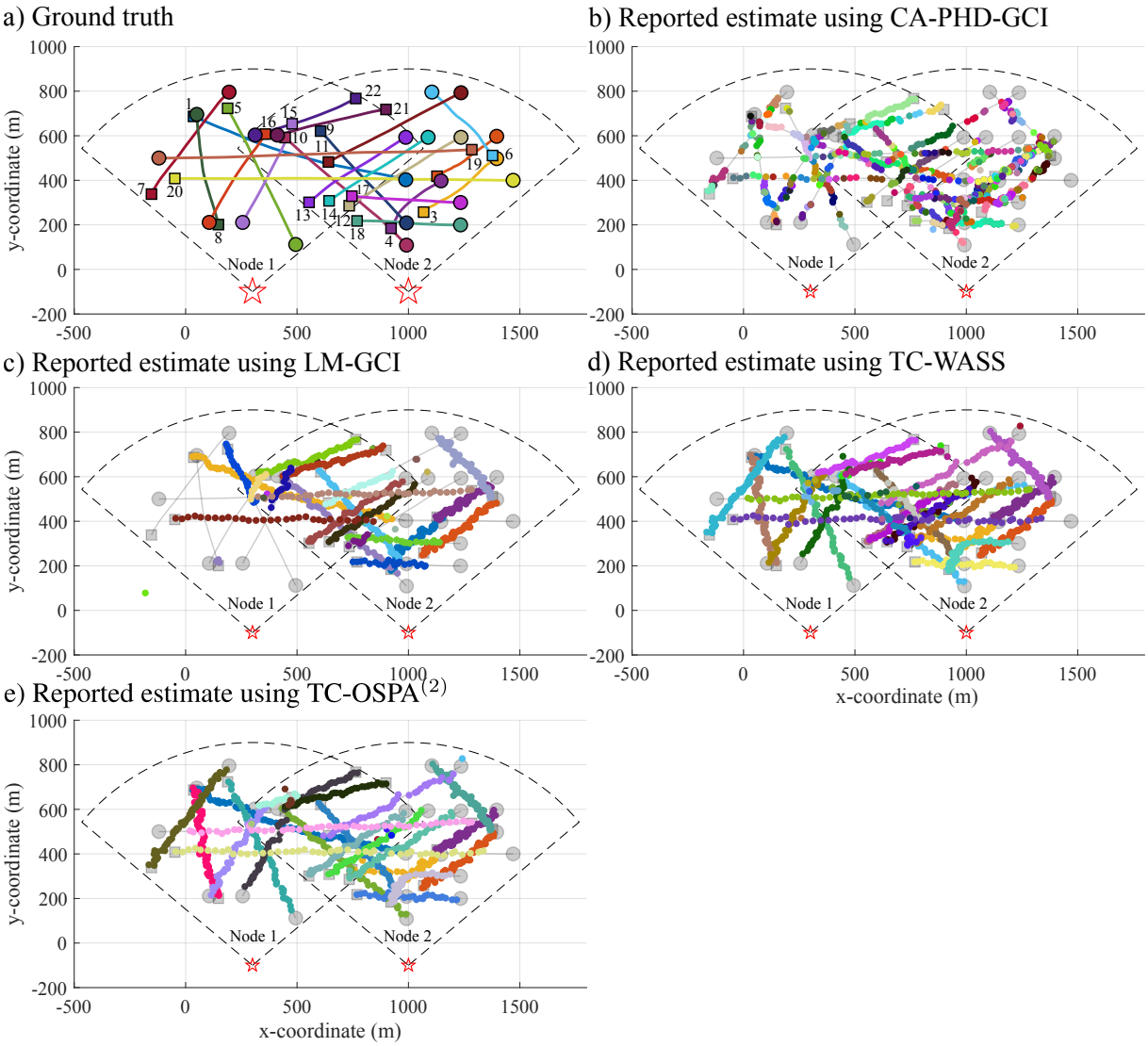


**Figure 6.4. Scenario 1 ground truth vs reported estimates at node 2.** a) ground truth settings, b) CA-PHD-GCI, c) LM-GCI, d) TC-WASS, e) TC-OSPA$^{(2)}$. Starting and stopping positions are denoted by ○ and □, respectively. Colour coding represents labels of the objects.

Fig. 6.4bcd depicts the reported estimate versus the ground truth tracks at node 2 of a particular trial using CA-PHD-GCI, LM-GCI, TC-WASS and TC-OSPA$^{(2)}$, respectively. The results confirm that TC-OSPA$^{(2)}$ can successfully detect and track all of objects without label inconsistency problems, regardless of whether these objects are in its FoV or not. In contrast, LM-GCI can only track most of the objects within its FoV. Although CA-PHD-GCI can detect all of objects, even outside of its FoV, it cannot determine whether these objects are the same or different objects because of the limitation nature of the PHD filter which can only track objects but not labels. Fig 6.5c shows the

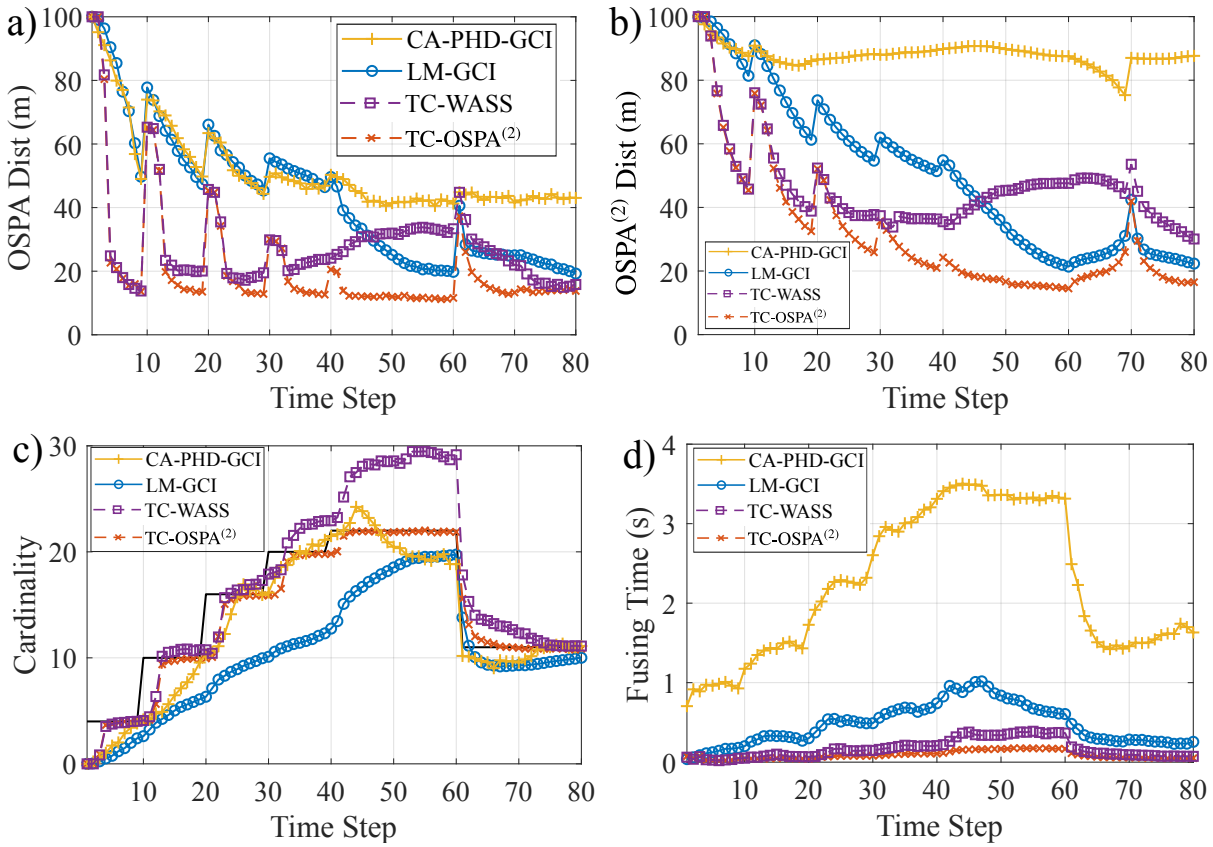


**Figure 6.5.** Comparison results for scenario 1: a) OSPA distance; b) OSPA$^{(2)}$ distance; c) Cardinality estimations; d) Fusing times.

cardinality estimation which further supports the above observation. It is expected that LM-GCI fails to detect all of the objects compared to CA-PHD-GCI, TC-WASS and TC-OSPA$^{(2)}$ since only the later three strategies are designed to cope with scenarios under limited FoV sensors.

Fig. 6.5abd and Table 6.1 present the performance comparison among three fusion strategies: CA-PHD-GCI, LM-GCI, TC-WASS and TC-OSPA$^{(2)}$ in terms of OSPA, OSPA$^{(2)}$ and fusing time over 100 Monte Carlo trials. It can be seen that TC-OSPA$^{(2)}$ outperforms other fusion strategies in large margins under the smallest fusing times. The reason is that TC-OSPA$^{(2)}$ has solved a challenging problem of label inconsistency for limited FoV sensors while fusing the best local estimates to reach consensus in both position and label estimations. The results also shows that TC-WASS fails to estimate the correct cardinality of truth objects since the Wasserstein metric can partially penalise the cardinality differences between tracks, as mentioned in the background section. As a result, TC-WASS is omitted in the next experiments.

**Table 6.1.** Comparison results of TC-OSPA$^{(2)}$ versus CA-PHD-GCI, LM-GCI, and TC-WASS in Scenario 1

| Strategies | OSPA (m) | OSPA$^{(2)}$ (m) | Fusing time (s) |
|---|---|---|---|
| CA-PHD-GCI | 51.6 | 87.9 | 2.216 |
| LM-GCI | 43.4 | 50.5 | 0.471 |
| TC-WASS | 29.0 | 46.3 | 0.168 |
| TC-OSPA$^{(2)}$ | **21.0** | **31.7** | **0.089** |

**Table 6.2.** Performance comparison for different $p_D$ values in Scenario 1

| | $p_D$ | CA-PHD-GCI | LM-GCI | TC-OSPA$^{(2)}$ |
|---|---|---|---|---|
| **OSPA (m)** | 0.7 | 78.9 | 61.1 | **41.3** |
| | 0.8 | 71.1 | 52.4 | **34.7** |
| | 0.9 | 64.4 | 44.7 | **29.0** |
| **OSPA$^{(2)}$ (m)** | 0.7 | 93.1 | 67.4 | **52.2** |
| | 0.8 | 90.6 | 59.6 | **45.8** |
| | 0.9 | 88.6 | 52.3 | **40.0** |

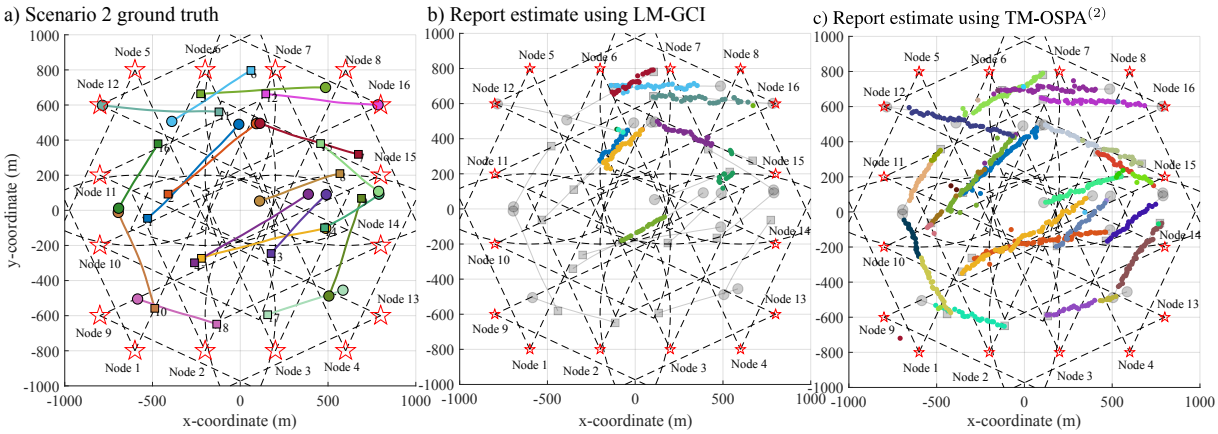


**Figure 6.6. Scenario 2 — ground truth vs reported estimates at node 7.** a) ground truth; b) LM-GCI; c) TC-OSPA$^{(2)}$ Starting and stopping positions are denoted by ○ and □, respectively. Colour coding represents labels of the objects.

Table 6.2 provides further performance comparison of TC-OSPA$^{(2)}$ versus LM-GCI and CA-PHD-GCI under different $p_D$ values. The results validate that although the tracking accuracy decreases when $p_D$ decreases as well as more label inconsistencies as observed in Remark 3, our proposed TC-OSPA$^{(2)}$ consistently outperforms the other two fusion strategies across different $p_D$ values.

**Table 6.3.** Comparison results of TC-OSPA$^{(2)}$ versus LM-GCI in Scenario 2

| Strategies | OSPA (m) | OSPA2 (m) | Fusing times (s) |
|---|---|---|---|
| LM-GCI | 78.8 | 81.6 | 1.48 |
| TC-OSPA2 | **24.6** | **45.9** | **0.32** |

### 6.4.2 Scenario 2 — A large number of nodes

To further demonstrate the effectiveness of our proposed fusion method, a scenario of 16 nodes with limited FoVs to track a time-varying and unknown number of mobile objects in a survey area of $[-1000, 1000]$ m $\times$ $[-1000, 1000]$ m is considered. The sensor has limited FoV that can only detection objects within its relative angle of the interval $[-25°, 25°]$ with $p_D = 0.98$ and $r_D = 1000$ m. There are up to 16 nodes positioned in the edge of the search area. The duration for this scenario is $K = 75$ s with various birth and death objects occur with the maximum number of objects is 18. The considered scenario settings is depicted in Fig. 6.6a. Note that for this scenario, we can only compare TC-OSPA$^{(2)}$ v.s. LM-GCI since it is unclear how CA-PHD-GCI is implemented for more than two nodes.

Fig. 6.6bc plots the estimate versus ground truth at node 7 using LM-GCI and TC-OSPA$^{(2)}$ respectively for one particular trial. Although unexpectedly, LM-GCI can detect and track a few objects outside of node 7's FoV, since LM-GCI is not designed to do that, there are remaining objects that LM-GCI cannot track. In contrast, TC-OSPA$^{(2)}$ can detect, track and assign correct labels for most of objects, regardless of the objects' locations. The results are further affirmed by cardinality estimation plotted in Fig. 6.7c which demonstrates that TC-OSPA$^{(2)}$ can detect all of 18 objects in this scenario, while LM-GCI can only averagely detect up to 4 objects over 100 MC runs.

Fig. 6.7abd provides additional comparison results between LM-GCI and TC-OSPA$^{(2)}$. The results further demonstrate the robustness of TC-OSPA$^{(2)}$ which significantly outperforms LM-GCI across three performance metrics: OSPA, OSPA$^{(2)}$ and fusing time. Table 6.3 provides detailed performance comparison results, which facilitates the effectiveness of our proposed fusion strategy for a distributed multi-sensor network.

Fig. 6.8 depicts the overall tracking performance for scenario 2 using TC-OSPA$^{(2)}$ at node 2 when the number of nodes is increased from 2 to 16. The results validate the scalability of our proposed fusion strategy, wherein the fusing time increases linearly to the number of nodes, *i.e.*, $\mathcal{O}(|\mathcal{S}|)$. Even when the number of nodes is 16, the fusing

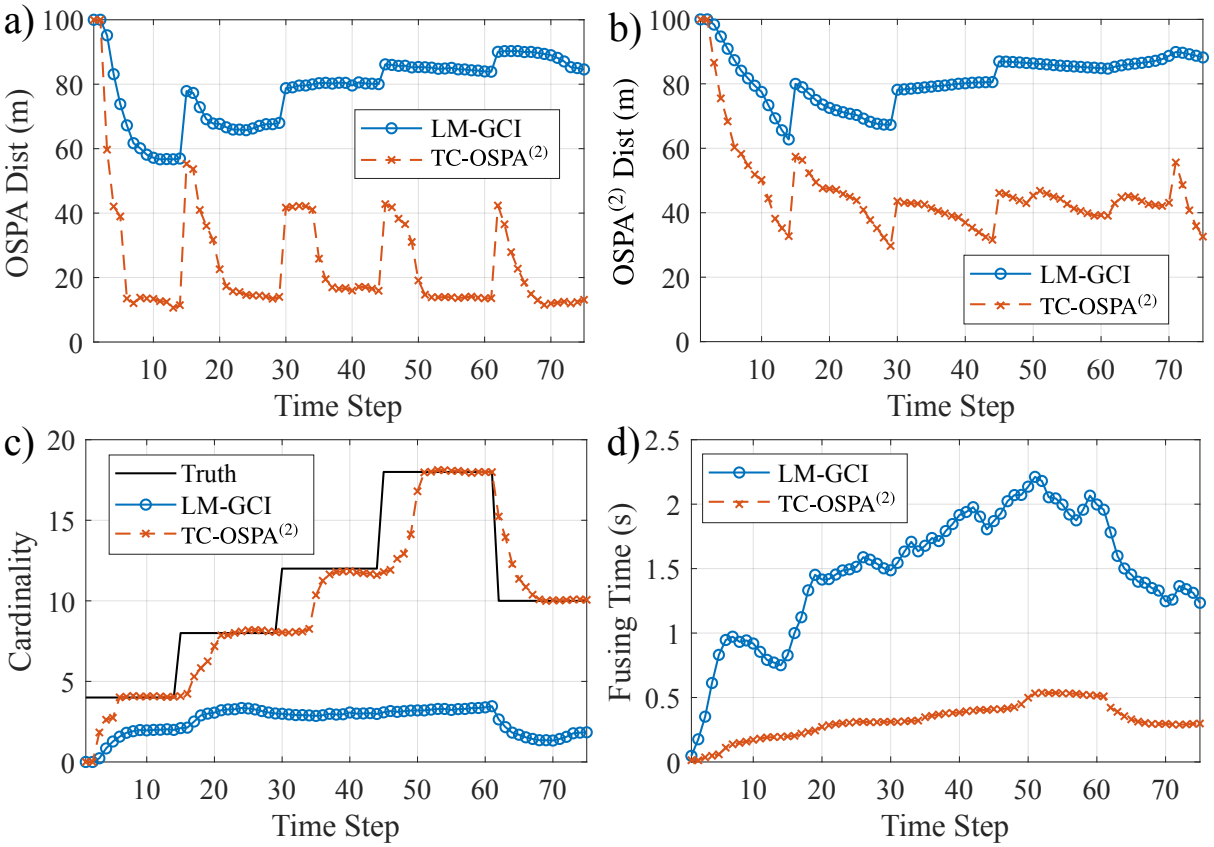


**Figure 6.7.** Comparison results at node 7 for scenario 2: a) OSPA distance; b) OSPA$^{(2)}$ distance; c) Cardinality estimations; d) Fusing times.

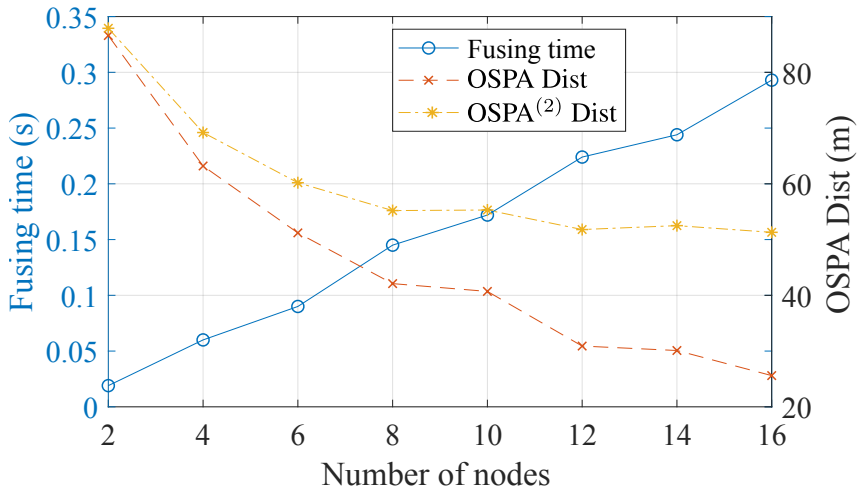


**Figure 6.8.** Tracking performance using TC-OSPA$^{(2)}$ at node 2 over 100 MC runs for scenario 2 when the number of nodes is increased from 2 to 16.

time is relatively small, which enables the real-time tracking in several applications. As expected, when the number of nodes increases, OSPA and OSPA$^{(2)}$ distances reduce

since the node can share its local estimate with other nodes to complement its limited FoV and improve its coverage area.

## 6.5 Conclusion

The chapter has provided new results of distributed multi-object tracking for multi-sensors with limited FoV sensors. A novel label consensus algorithm coupling with a track consensus method based on OSPA$^{(2)}$ has been devised for fully distributed and scalable fusion of information from distributed multi-sensor networks. The experimental results demonstrate the effectiveness of our approach compared to GCI-based fusion methods. Future work should consider incorporating planning algorithms jointly with the proposed fusion methods to empower the autonomy of distributed systems in tracking multi-object.

# Chapter 7

# Conclusion

THIS chapter concludes the dissertation and suggests directions for future work.

## 7.1 Summary

This dissertation presents novel approaches for achieving autonomy for single or multiple UAVs to search and track multiple mobile objects of interest. Our formulation is built on the RFS-(M)POMDP framework and addresses all of the challenges of MOT and online planning problems. In particular, the MOT problem is formulated under the RFS framework to address MOT challenges such as false-alarms, misdetections, unknown data associations and the time-varying number of objects. For online path planning problems, we adopt POMDP for a single UAV and the centralised MPOMDP for multiple UAVs to compute optimal control actions in an online manner. The information-based reward functions have been implemented to calculate optimal actions by maximising information gain to reduce overall estimation uncertainty since more information naturally implies less uncertainty. Our formulation yields a value function as a submodular set function, which allows us to apply the greedy algorithm for computing optimal control actions for multi-agent planning in real-time with a lower bound performance guarantee. Besides that, a distributed fusion algorithm has been developed for a network of static agents with limited FoV sensors, which is an essential building block for multi-agent path planning for MOT in a distributed manner. Our proposed approaches work well in multiple situations compared to state-of-the-art methods, regardless of the number of agents or the sensor FoV sizes.

Chapter 3 presents a real-world autonomous aerial robotic system that is able to search, localise and track multiple mobile VHF radio-tagged objects under noisy RSSI-based measurements. The formulation is based on using a particle filter for MOT and the POMDP for path planning. The Rényi divergence information-based reward function is used to compute the optimal control actions. The field experimental results demonstrated our proposed approach which can track multiple wildlife collars accurately. Additionally, our proposed sensor system is lightweight, which helps reduce the overall UAV's payload and improve flight times.

In Chapter 4, an online path planning algorithm of a single UAV for joint detection and tracking of multiple radio-tagged objects under low SNR environments has been proposed. We use a JMS TBD-LMB filter to track multiple objects, while the POMDP framework controls the UAV. Two information-based reward functions using Rényi divergence and Cauchy-Schwarz divergence are investigated to compute the optimal control actions. A measurement likelihood function is proved to be separable in practice, which results in an efficient TBD-LMB filter to track multiple radio-tagged

objects under low SNR conditions. Simulation results validated our proposed method, which significantly reduces estimation errors compared to traditional DTT methods or tracking without planning, especially under low SNR environments.

Chapter 5 extends the RFS-POMDP framework for a single agent to RFS-MPOMDP framework for multi-agents. A multi-objective planning for multi-agents to search and track an unknown and time-varying number of mobile objects is formulated. We proved that our formulation results in a value function that is monotone and submodular. Experimental results confirm the capability of our approach, which also validate the lower bound performance guarantee of using the greedy algorithm to compute control actions for multiple agents in real-time.

In Chapter 6, a fast and efficient distributed MOT algorithm for multi-agents with limited FoV sensors is developed. The proposed fusion strategy operates on local multi-object trajectories instead of multi-object densities, which results in real-time tracking under limited computational resources. We use the OSPA$^{(2)}$ to estimate the similarities among trajectories for achieving track consensus. Furthermore, a novel label consensus algorithm is devised to minimise label inconsistency problems due to limited FoV sensors. Experimental results confirm that our proposed approach significantly outperforms GCI-based fusion methods in both speed and accuracy.

## 7.2 Future Work

There are a number of possible areas worthy of further investigation to improve the autonomy of multiple UAVs under RFS-POMDP framework, including *i)* multi-UAV systems designs and *ii)* algorithmic developments.

### 7.2.1 Multi-UAV systems designs

In Chapter 3, while the SDR devices may be replaced to achieve a greater detection range, as we discussed in Section 3.7.2, future work should focus on the development of new antenna designs. We designed, simulated and built a compact, folded two element Yagi antenna. Further research efforts to investigate antenna design techniques can lead to lightweight higher gain antennas to increase the detection range and survey area.

Furthermore, the range of the 2.4 GHz wireless link we employed for communicating between the GCS and the UAV has limited outdoor range—see Figure 3.2. Although this is not a problem given the limited test site available for our work, building a practical tool requires addressing this potential problem. Thus, future work should piggyback data on the telemetry channel using the long-range 915 MHz radio channel (VonEhr et al., 2016). Alternatively, the GCS may be removed from the loop, by embedding all of the tracking and planning algorithm on the UAV itself using an embedded computing platform (*e.g.*, Odroid XU4, Mobile GPU), to increase the system reliability and search area, by eliminating the transmission power consumed by the additional 2.4 GHz radio channel.

Although it is possible to perform 3D tracking exploiting the simplicity of range only measurements as in Appendix B, the next step is to build a sophisticated range-only measurement model to account for complex signal characteristics in 3D environments, and to validate it in field experiments. Additionally, the practical challenge is that we need to obtain accurate UAV altitude measurements to implement a robust 3D tracking formulation. Commercial off the shelf UAVs such as the 3DR IRIS+ that we used for building our autonomous system employ a barometer to determine height. We observed in flight tests that the height measurement is unreliable, fluctuates over time and often depends on weather conditions; as also observed in (Szafranski et al., 2013; Liu et al., 2014). Thus, we leave it for future work to address the problem of accurately estimating the altitude of a UAV. Two approaches that can be considered include: *i)* filtering the barometer sensor data using, for example, a Kalman filter (Liu et al., 2014); and *ii)* the use of a LiDAR sensor or a radar-based sensor for more accurate height above ground estimations (Schartel et al., 2018). Alternatively, employing the existing implementation on all topographical conditions require a UAV capability to maintain a fixed relative altitude above ground.

Another interesting area worthy of exploration is developing control actions for hybrid vertical take-off and landing (VTOL) fixed-wing UAVs to extend flight times and cover longer distances (Unmanned Systems Technology, 2019). Although hybrid VTOL UAVs can hover, its heading (or so-called yaw) is significantly harder to control than the investigated multi-copter drones. For RSSI-based measurements, the received signal strength depends heavily on UAV's heading because the antenna gain is typically directional. The reason is that directional antennas help to improve detection range as well as identify the angle of arrival. Therefore, the planning algorithm needs

to take into account the aerial dynamic behaviours of hybrid VTOL UAVs and the antenna gain when calculating the control actions to achieve optimal results.

In Chapter 4, an efficient TBD-LMB filter has been derived. A next natural step is to implement the proposed TBD-LMB filter with an information-based reward function on the UAV itself to fully explore its efficiency under low SNR conditions. However, the complexity of processing the raw received measurements through STFT algorithms is quite high, which prevents using a standard computing unit such as Raspberry Pi or Intel Edison, as implemented in Chapter 3. One of potential companion computers would be NVIDIA Jetson TX2 Module, wherein the tracking and planning algorithms could be computed efficiently using the embedded GPUs. Additionally, the optimisation of the proposed TBD-LMB algorithm could be explored to further improve the computing time.

### 7.2.2 Algorithmic developments

For multi-agent planning, a common approach is to use a centralised MPOMDP (Messias, Spaan and Lima, 2011) wherein a centralised controller calculates optimal control actions and sends it to all agents, as presented in Chapter 5. However, the centralised approach is limited in scalability. Factored-POMDP (Oliehoek et al., 2008) can be employed to achieve further system scalability but centralised methods are prone to single point of failure and require reliable and fast communications between agents and the centralised controller. It is still extremely challenging to plan and track mobile objects in an online manner without any communications among agents as in decentralised POMDP (Dec-POMDP), in which its exact solution are NEXP-complete (Bernstein et al., 2002). A more reliable approach could be using distributed POMDPs, such as networked distributed POMDP (ND-POMDP) (Nair et al., 2005) to exploit neighbouring interactions to coordinate among agents for planning under uncertainty using distributed constraint optimisation. This approach also naturally fits with our proposed distributed MOT algorithms in Chapter 6. However, achieving a global goal for multiple agents to track multiple mobile objects in distributed settings is an NEXP-complete problem in the worst case scenario (Rizk, Awad and Tunstel, 2018). Future research directions can explore new planning strategies to the problem of distributed control of a team of UAVs for searching and tracking. Moreover, since the observed signals have built-in object labels, the problem considered in Chapter 5 is less complex than the case where the objects have no labels.

Thus, one possible research direction is to generalise the proposed framework in Chapter 5 for the problem where the data associations are unknown using the novel MS-GLMB filter recently proposed in (Vo, Vo and Beard, 2019).

In this dissertation, we assumed that the locations of UAVs can be obtained accurately using the global position system (GPS). It could be useful to consider scenarios wherein GPS information is not available or not reliable such as in GPS-denied environments (Lange, Sunderhauf and Protzel, 2009; Song et al., 2015). One of feasible approaches is building 3D simultaneous localisation and mapping (SLAM) environments, such as in (Artieda et al., 2009) for UAVs or using a laser range finder (Fossel et al., 2013; Song et al., 2015). For multi-agent SLAM, a centralised approach has been investigated in (Schmuck and Chli, 2017). Given our interest in multi-agent planning for MOT, it is still an open question for tracking multiple objects in a distributed manner when the locations of UAVs are unknown.

# Appendix A

# Pseudo-codes for Distributed Algorithms of Chapter 6

The following pseudo-codes provide detailed implementation of Algorithm 6.A.1 of determining matched pairs based on OSPA$^{(2)}$ distance between two tracks, Algorithm 6.A.2 of updating association history for these matched pairs, Algorithm 6.A.3 for updating the preserved labels of preserved estimates, Algorithm 6.A.4 for updating reported labels to minimise label inconsistency problems.

**Algorithm 6.A.1** DetermineMatchedPairs

**Input**: $\mathbf{X}_{j:k}^{(a)}; \mathbf{X}_{j:k}^{(b)}; L_k^{(a)}; L_k^{(b)}$
**Output**: $Q_k = [Q_k^{(a)}, Q_k^{(b)}]$
$C := \mathsf{zeros}(|L_k^{(a)}|, |L_k^{(b)}|)$;
**for** $m = 1 : |L_k^{(a)}|$ **do**
    $l^{(m)} := L_k^{(a)}(:, m)$;
    **for** $n = 1 : |L_k^{(b)}|$ **do**
        $l'^{(n)} := L_k^{(b)}(:, n)$;
        $C_{m,n} := \tilde{d}^{(c)}(\mathbf{x}^{(l_m)}, \mathbf{x}^{(l'_n)})$ via (6.4)
    **end for**
**end for**
$S := \mathsf{RankedAssignment}(C)$; ▷ using Murty's algorithm.
$S := S \odot (C < \mathcal{C}_{\text{match}})$; ▷ assignments with cost less than $\mathcal{C}_{\text{match}}$ only.
$i_k^{(a)} := [1 : |L_k^{(a)}|]^T$; $i_k^{(b)} := [1 : |L_k^{(b)}|]^T$;
$Q_k^{(a)} := i_k^{(a)}$; $Q_k^{(b)} := S \cdot i_k^{(b)}$; $Q_k := [Q_k^{(a)}, Q_k^{(b)}]$;
$Q^{\text{check}} := [Q_k^{(a)} \odot Q_k^{(b)}] > 0$; ▷ ensure $S_{m,n} = 1$.
$Q_k := Q_k(:, Q^{\text{check}})$;

**Algorithm 6.A.2** UpdateAssociationHistory

**Input**: $\Xi_{1:k-1}^{(a,b)}$; $L_{1:k}^{(a)}$; $L_{1:k}^{(b)}$; $Q_k = [Q_k^{(a)}, Q_k^{(b)}]$
**Output**: $\Xi_{1:k}^{(a,b)}$
$\Xi_{1:k}^{(a,b)} := \mathsf{zeros}(|L_{1:k}^{(a)}|, |L_{1:k}^{(b)}|)$;
$\Xi_{1:k}^{(a,b)}(1 : |L_{1:k-1}^{(a)}|, 1 : |L_{1:k-1}^{(b)}|) := \Xi_{1:k-1}^{(a,b)}$;
$i_{1:k}^{(a)} := 1 : |L_{1:k}^{(a)}|$; $i_{1:k}^{(b)} := 1 : |L_{1:k}^{(b)}|$
**for** $i = 1 : |Q_k^{(a)}|$ **do**
    $m := Q_k^{(a)}(i)$; $n := Q_k^{(b)}(i)$;
    $l^{(m)} := L_k^{(a)}(: m)$; $l'^{(n)} := L_k^{(b)}(:, n)$;
    $i^{(m)} := i_{1:k}^{(a)}(l^{(m)} = L_{1:k}^{(a)})$;
    $i^{(n)} := i_{1:k}^{(b)}(l'^{(n)} = L_{1:k}^{(b)})$;
    $\Xi_{1:k}^{(a,b)}(i^{(m)}, i^{(n)}) := \Xi_{1:k}^{(a,b)}(i^{(m)}, i^{(n)}) + 1$;
**end for**

**Algorithm 6.A.3** UpdatePreservedLabels

**Input**: $\Xi_{1:k}^{(a,b)}$; $L_{1:k}^{(a)}$; $L_{1:k}^{(b)}$; $L_k^{\text{pres}}$; $\bar{L}_k$
**Output**: $L_k^{\text{pres}}$
$i_{1:k}^{(a)} := 1 : |L_{1:k}^{(a)}|$; $i_{1:k}^{(b)} := 1 : |L_{1:k}^{(b)}|$
**for** $i = 1 : |L_k^{\text{pres}}|$ **do**
    **if** $L_k^{\text{pres}}(:, i) \in L_{1:k}^{(b)}$ **then** ▷ only update labels from node $b$.
        $l' := L_k^{\text{pres}}(:, i)$;
        $n := i_{1:k}^{(b)}(l' = L_{1:k}^{(b)})$;
        $\xi^{(a,l)} := \Xi_{1:k}^{(a,b)}(:, n)$;
        count:= $|\xi^{(a,l)}(\xi^{(a,l)} > 0)|$;
        **while** count $> 0$ **do**
            count :=count$-1$;
            $[\sim, m] :=$ max$(\xi^{(a,l)})$; ▷ pick highest value index.
            $l := L_{1:k}^{(a)}(:, m)$;
            **if** $l \notin \left[L_k^{\text{pres}} \cup \bar{L}_k\right]$ **then** ▷ ensure labels' uniqueness.
                $L_k^{\text{pres}}(:, i) := l$; ▷ update the label.
                break; ▷ escape while loop.
            **end if**
            $\xi^{(a,l)}(m) := 0$;
        **end while**
    **end if**
**end for**

**Algorithm 6.A.4** UpdateReportedLabels

**Input**: $\Xi_{1:k}^{(a,b)}$; $L_{1:k}^{(a)}$; $L_{1:k}^{(b)}$; $L_k^{\text{rep}}$
**Output**: $L_k^{\text{rep}}$
$i_{1:k}^{(a)} := 1 : |L_{1:k}^{(a)}|$; $i_{1:k}^{(b)} := 1 : |L_{1:k}^{(b)}|$
**for** $i = 1 : |L_k^{\text{rep}}|$ **do**
    **if** $L_k^{\text{rep}}(:, i) \in L_{1:k}^{(a)}$ **then** ▷ update labels from node $a$.
        $l := L_k^{\text{rep}}(:, i)$;
        $m := i_{1:k}^{(a)}(l = L_{1:k}^{(a)})$;
        $\xi^{(b)} := \Xi_{1:k}^{(a,b)}(m, :)$;
        count:= $|\xi^{(b)}(\xi^{(b)} > 0)|$;
        **while** count $> 0$ **do**
            count :=count$-1$;
            $[\sim, n] :=$ max$(\xi^{(b)})$; ▷ index highest value.
            $l' = L_{1:k}^{(b)}(:, n)$;
            **if** $l' \notin L_k^{\text{rep}}$ and $l'(1) < l(1)$ **then** ▷ ensure labels' uniqueness and only update new labels.
                $L_k^{\text{rep}}(:, i) := l'$; ▷ update the label.
                break; ▷ escape while loop.
            **end if**
            $\xi^{(b)}(n) := 0$;
        **end while**
    **end if**
**end for**

# Appendix B

# Software in the Loop Study for Locating Radio-tags in a 3D Space

Our problem formulation in Chapter 3 developed a tracking and planning problem for a three dimensional space (3D). However, our simulations and experiments were limited to 2D spaces where the terrain was relatively flat and the height of the UAV above ground was constant. Here, we investigate the implementation of the 3D tracking and planning problem to validate the capability of the range-only method proposed in Chapter 3 to track multiple mobile radio tags under real-world digital terrain models in hilly terrains in a simulated software-in-the-loop environment of a quad-copter.

## B.1 Motivation and Contribution

In this appendix, we investigate our tracking and planning formulation in Chapter 3 in 3D to account for unknown terrains, especially in hilly areas. Although Chapter 3 has demonstrated an aerial robot system, for the first time, capable of planning trajectories to track and localise multiple mobile VHF wildlife radio tag objects, the validation of the method in a 3D environment remains.

As in Chapter 3, we use received signal strength indicator (RSSI) measurements, which exploits the simplicity of antenna and receiver designs to build a lightweight payload system, to validate our approach in a 3D environment. Notably, all of existing UAV-based methods assume that the terrains are flat and implemented tracking and planning algorithms to localise radio-tagged objects in two dimensions (latitudes and longitudes only) on the ground plane. Elevations (the ground surfaces) have been largely ignored in the previous work. This additional study takes the first step towards autonomous tracking and localising under unknown terrains in 3D environments using a UAV with RSSI-based measurements.

In summary, the key study contributions are:

- Implementing a 3D tracking and planning formulation using RSSI-based methods in a software-in-the-loop (SITL) simulation of a quad-copter,
- Simulating tracking and localising multiple mobile radio-tagged objects in hills or valleys where the terrain information is unknown,
- Comparison of the developed 3D tracking method to track and localise radio-tags in unknown terrains with one where the terrain information is known. Here, the investigations are based on the real-world digital elevation model (DEM) data published by (Australia-Geoscience, 2018) for a simulated SITL quad-copter.

## B.2 Problem Statement

We consider the problem of tracking and localising multiple mobile radio-tagged objects in the hilly terrains using a UAV. The proposed platform is discussed in Chapter 3, with the following elements:

- A civilian, commercial and low-cost UAV with a accurate global positioning system (GPS) measurements in latitude and longitude, but using an unreliable barometer sensor in altitude measurements. The UAV manoeuvrability is determined by that of a quad-copter.
- A sensor system—the main payload—includes a directional VHF antenna to receive the transmitted signals, an embedded computer module connected to a software-defined radio device to detect and measure the received signal strength indicator (RSSI) through VHF antenna.

Further, we assume that each radio-tag transmits an on-off-keying signal with known transmission power $P_0$ in every $T_0$ seconds. The object is located in a hilly area where its altitude can vary in $[z_{\min}, z_{\max}]$ m. We did not consider the exploration problems in this work where the reward functions can be formulated in both exploration and localisation parameters (Charrow, Michael and Kumar, 2015). Instead, we assume that the UAV can detect all of the objects, which is reasonable in a moderate size search area; we concentrate on improving the tracking performance for detected objects.

## B.3 Problem Formulation

In this work, we focus on formulating the problem of tracking and localising radio-tagged objects in unknown terrains and follow our previous work in Chapter 3. The state of a single object is $x = [p_{\mathrm{x}}^{(x)}, p_{\mathrm{y}}^{(x)}, p_{\mathrm{z}}^{(x)}]^T \in \mathbb{R}^3$, which is the object 3D position in x, y and z axes of the Cartesian coordinate system. The state of a UAV is $u = [p^{(u)}, \theta^{(u)}]^T \in \mathbb{R}^3 \times [0, 2\pi)$, where $p^{(u)} = [p_{\mathrm{x}}^{(u)}, p_{\mathrm{y}}^{(u)}, p_{\mathrm{z}}^{(u)}]^T \in \mathbb{R}^3$ is the UAV position in 3D coordinate; $\theta^{(u)}$ is the UAV heading. Further, we assume that the number of objects $N_t$ in the search area is known, and the search operation terminates when all of the searching objects are tracked and localised.

### B.3.1 Multi-object tracking

We propose using a particle filter to implement our tracking algorithm to account for the non-linear system dynamics and noisy measurement data from signal strength measurements interfered with by radio-wave scattering and attenuation or thermal noise of the receiver (Nguyen et al., 2019a). Since each object is uniquely identified

by its frequency, the RSSI-based measurements provide a known data association. Further, we assume that there are no false-alarms or misdetections for our RSSI-based measurements as in (Cliff et al., 2015; Nguyen et al., 2018a, 2019a). Therefore, we can track and localise multiple radio-tagged objects by running multiple particle filters simultaneously, one particle filter for *each object*, as proposed in (Charrow, Michael and Kumar, 2015; Nguyen et al., 2019a). The particle filter requires correctly modelling for both object transition and observation models to achieve good performance.

**Object transition model:** For wildlife objects, their dynamic behaviours are usually unpredictable, thus we model their behaviours as a random walk model, *i.e.*,

$$f_{k|k-1}(x_k|x_{k-1}) = \mathcal{N}(x_k; x_{k-1}, Q^{(x)}), \tag{B.1}$$

where $\mathcal{N}(\cdot; \mu, Q)$ denotes a Gaussian density with mean $\mu$ and co-variance $Q$; $Q^{(x)} = [\sigma_x^2, \sigma_y^2, \sigma_z^2] I_3$ is the $3 \times 3$ co-variance matrix of the process noise, and $I_n$ denotes the $n \times n$ identity matrix.

**Observation model:** We consider the *LogPath* measurement model experimentally validated with VHF frequencies in Chapter 3. Here, the received power $h(x_k, u_k)$ [dBm] at the UAV with state $u_k$ transmitted from object with state $x_k$ comprises only the LOS component,*i.e.*,

$$h(x_k, u_k) = P_0 - 10n \log(d(x_k, u_k)) + G_r(x_k, u_k). \tag{B.2}$$

Here, $P_0$ is the reference power [dBm]; $n$ is the unit-less path loss constant, which characterises how signal attenuates over the distance with a typical range from 2 to 4; $d(x_k, u_k) = ||x_k - p^{(u_k)}||$ is the distance between the object and the UAV; $G_r(x_k, u_k)$ is the directional antenna gain, which depends on the UAV heading $\theta^{(u_k)}$ and its relative position to the object $x_k$.

The measured power or the received signal strength indicator (RSSI) $z_k$ [dBm] is corrupted with noise, *e.g.*, thermal noise or signal interference from other sources. We assume the noise is white, thus, the measurement likelihood model is

$$g_k(z_k|x_k) = \mathcal{N}(z_k; h(x_k, u_k), Q^{(z)}), \tag{B.3}$$

where $Q^{(z)}$ is the $1 \times 1$ co-variance matrix of the measurement noise.

### B.3.2 Path planning using the Shannon entropy information gain

In this section, we present our approach to calculate an optimal control action for the UAV. At time $k$, the UAV needs to plan how it will navigate over the time interval $k+1 : k+H$ with the look-ahead horizon $H$. Since there are multiple objects in the search area, we select the object with the strongest RSSI-based measurement as the one to be tracked and localised first (Nguyen et al., 2019a). Formally, suppose $Z_k(X_k) = \bigcup_{x \in X_k} z_k(x)$ is a set of measurements at time $k$ generated from the respective set of objects $X_k = \{x_k^{(1)}, \ldots, x_k^{(N_t)}\}$, and $F_k$ is of the set of localised objects (an object is considered localised if its estimation uncertainty is smaller than a predefined bound), the selected object $\overset{*}{x}_k$ for the path planning at time $k$ is given by,

$$\overset{*}{x}_k = \underset{x \in X_k \setminus F_k}{\arg\max}\, Z_k(X_k). \tag{B.4}$$

Let $\mathcal{A}_k$ be a discrete set of control actions for the UAV at time $k$. We define $\mathcal{A}_k$ contains $|\mathcal{A}_k|$ number of actions that control the UAV to change its heading to one of the following $\{0, 2\pi/|\mathcal{A}_k|, \ldots, 2\pi(1-1/|\mathcal{A}_k|)\}$ angles, then moves forward according to the selected angle until another control action applies. For each control action $a \in \mathcal{A}_k$ applies to the UAV, it generates a discrete sequence of the UAV poses $u_{k+1:k+H}(a) = [u_{k+1}, \ldots, u_{k+H}]$ with corresponding measurements $z_{k+1:k+H}(a) = [z_{k+1}, \ldots, z_{k+H}]$.

The goal in path planning is to find an optimal control action $\overset{*}{a}_k \in \mathcal{A}_k$ that maximises the expected reward, *i.e.*,

$$\overset{*}{a}_k = \underset{a \in \mathcal{A}_k}{\arg\max}\, \mathbb{E}\big[\mathcal{R}_{k+H}(a)\big]. \tag{B.5}$$

Since the expected reward requires an integration, which does have an analytic formula, we implement the Monte Carlo integration (Ristic and Vo, 2010; Beard et al., 2017; Nguyen et al., 2019a) by drawing multiple sampled measurements $z_{k+1:k+H}^{(m)}(a)$ for $m = 1, \ldots, M$, then calculate the sampled reward $\mathcal{R}_{k+H}^{(m)}(a)$. Thus, the expected reward can be approximated by the mean of all the sampled rewards, *i.e.*,

$$\mathbb{E}\big[\mathcal{R}_{k+H}(a)\big] \approx \frac{1}{M} \sum_{m=1}^{M} \mathcal{R}_{k+H}^{(m)}(a). \tag{B.6}$$

In this work, we implement the change in Shannon entropy as the reward function[14] as in (Cliff et al., 2015; Charrow, Michael and Kumar, 2015):

$$\mathcal{R}^{(m)}_{k+H}(a) = \mathcal{H}(\pi_{k+H|k}(\overset{*}{x}_k|z_{1:k})) - \mathcal{H}(\pi_{k+H}(\overset{*}{x}_k|z_{1:k}), z^{(m)}_{k+1:k+H}(a)). \tag{B.7}$$

For notational simplicity, let $\pi_1 \triangleq \pi_{k+H|k}(\overset{*}{x}_k|z_{1:k})$ and $\pi_2 \triangleq \pi_{k+H}(\overset{*}{x}_k|z_{1:k}, z^{(m)}_{k+1:k+H}(a))$. Since we use the particle filter as our tracking filter, each density can be approximated by the same set of particles with different weights:

$$\pi_1 \approx \{(w^{(i)}_1, \tilde{x}^{(i)})\}^{N_s}_{i=1}; \pi_2 \approx \{(w^{(i)}_2, \tilde{x}^{(i)})\}^{N_s}_{i=1}. \tag{B.8}$$

Thus, the reward function in (B.7) can be approximated as followed:

$$\mathcal{R}^{(m)}_{k+H}(a) \approx \sum^{N_s}_{i=1} \left[w^{(i)}_2 \log(w^{(i)}_2) - w^{(i)}_1 \log(w^{(i)}_1)\right]. \tag{B.9}$$

## B.4 Software In The Loop Experiments

In this section, we validate and demonstrate our approach by tracking and localising multiple radio-tagged objects in two different unknown terrains. Further, we compare our 3D tracking algorithm with a tracking method where the terrain information is already known. The terrain information is based on the real-world DEM data published by (Australia-Geoscience, 2018) with 5 m in latitude and longitude resolutions, and $\pm$0.3 m in altitude errors.

### B.4.1 Simulation experimental setup

We evaluate our algorithm using the real-time emulated SITL environments as shown in Figure B.1. The tracking and planning algorithm is written in MATLAB, which sends control actions in way-points through the *Telemetry Host Tool* and the Input/output proxy—*IO proxy*, both are written in Rust, to the *DroneKit-SITL* simulator (Ryan et al., 2015) using the MAVLink protocol. For the *DroneKit-SITL*, we use the *copter-3.3* library to emulate a quad-copter. Further, the *QGroundControl* (a popular and cross-platform

[14]Notably, multiple other information gain measures can be employed. In Chapter 3, we investigated several reward functions. We selected Shannon entropy here due to its simplicity and because our goal is to take the first steps to demonstrate that RSSI based measurements from an aerial robot can be used to realise tracking in realistic 3D settings.

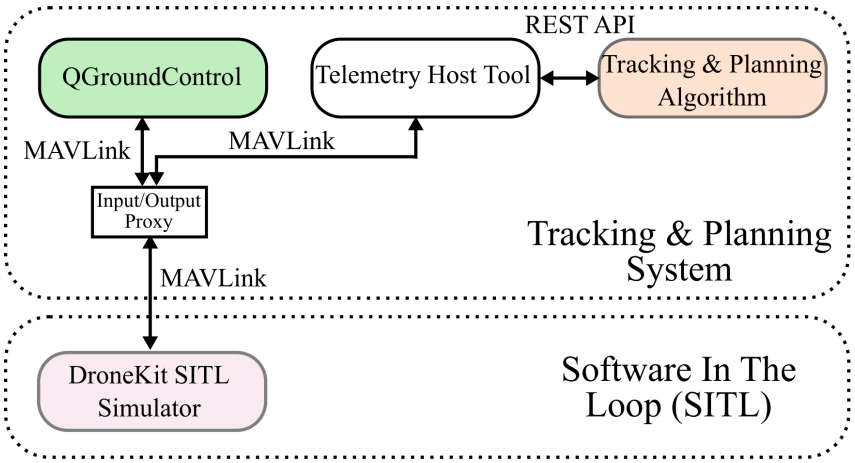


**Figure B.1. Block diagram of our propose SITL settings for emulated experiments..** The *Tracking and Planning System* communicates with the *DroneKit SITL* simulator through the MAVLink protocol.

ground station control software) can also communicate to the *DroneKit-SITL* simulator to facilitate and control the emulated copter in arming, taking off, and changing its altitude to a defined altitude above ground level (AGL). The tools and software developed for the **TrackerBots** project will be publicly available at our project repository[15]

We conduct several software-in-the-loop (SITL) trials under two different terrain settings: i) South Australia (SA) - Lower Glenelg National Park; ii) New South Wales (NSW) - Dorrigo National Park as shown in Figure B.2 to verify and demonstrate the capability of planning to track multiple mobile objects with RSSI based measurements from an aerial robot.

### B.4.2 Algorithm evaluations:

To evaluate our proposed algorithm, we measure the Root Mean Square (RMS) error—the average error distance between the objects' estimated locations versus its ground truths—RMS $= \sum_{i=1}^{N_t} ||x_{\text{truth}}^{(i)} - x_{\text{est}}^{(i)}|| / N_t$ [m], and the flight time [s]—the time a UAV takes to localise all of the objects, including planning time. As in Chapter 3, an object is considered tracked and localised if its estimation uncertainty is smaller than the predefined bound: 15 m for the x-axis and y-axis, and 25 m for the z-axis. The reason z-axis has a higher bound is because the directional antenna does not provide

[15]The TrackerBots project repository https://github.com/AdelaideAuto-IDLab/TrackerBots

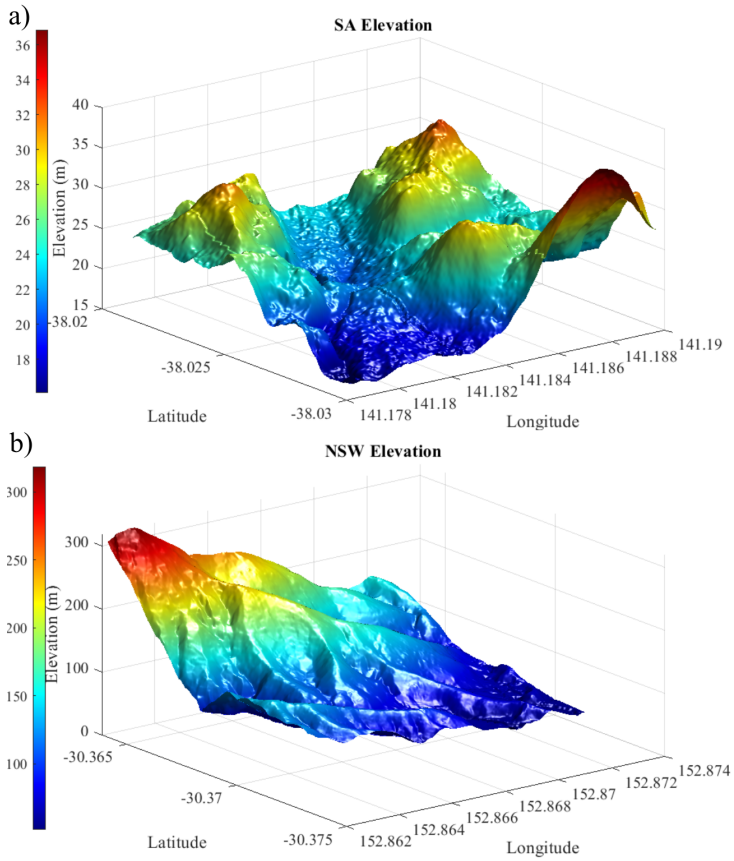


**Figure B.2. The terrain information for two site settings.** a) Lower Glenelg National Park terrain, South Australia (SA); b) Dorrigo National Park terrain, New South Wales (NSW).

an accurate antenna gain in z-axis causing higher uncertainty in the estimation (see the antenna pattern modelled and evaluated in Chapter 3 where the measurements validated the pattern in the $xy$ plane due to the difficulty of accurately controlling the UAV position to measure the field pattern in the $xz$ plane).

### B.4.3 Scenario 1:

The first scenario considers tracking and localising three mobile radio-tagged wildlife in Lower Glenelg National Park, South Australia (SA). We selected a search area of 1000 m $\times$ 1000 m (100 hectares) where the elevation changes from 16 m to 36 m based on the real-world digital elevation model (DEM) from (Australia-Geoscience, 2018), as shown in Figure B.2a. Its initial position in latitude, longitude, elevation is $[-38.0300, 141.1783, 17.7]^T$, which is converted to $[0, 0, 17.7]^T$ m in the xyz-axes.

For generating the ground truth, the initial positions of three mobile objects are $[320, 361, 21.5]^T$ m, $[826, 640, 26.7]^T$ m and $[166, 796, 30.3]^T$ m. objects are assumed to follow the random walk model with standard deviations in x-axis and y-axis as $\sigma_x^{(x)} = \sigma_y^{(x)} = 1$ (m/s), while its elevation in z-axis is derived from the DEM data based on its x and y positions.

For tracking and planning algorithm *without terrain awareness*, the object location is unknown and its initial distribution is sampled from a uniform distribution over the predefined ranges with the number of particles $N = 40,000$, *i.e.*,

$$\pi_0(x_0) = \mathcal{U}[0, 1000] \times \mathcal{U}[0, 1000] \times \mathcal{U}[12.7, 37.7],$$

where $\mathcal{U}[a, b]$ denotes the uniform distribution on the interval $[a, b]$ (m). The co-variance matrix of the process noise is $Q^{(x)} = [1, 1, 0.1]^T I_3$ $(\text{m/s})^2$. We set the measurement duration $T_0 = 1$ s, the measurement noise $Q^{(z)} = 5^2$ $(\text{dBm})^2$, the reference power $P_0 = -35.4$ dBm, the path loss constant $n = 2$, and the look-a-head horizon time step $H = 10$. The UAV is armed, taken off and its altitude is set to 80 m AGL using QGroundControl, *i.e.*, its initial state is set at $u_0 = [10 \text{ m}, 10 \text{ m}, 97.7 \text{ m}, \pi/4 \text{ rad}]^T$ and its maximum ground speed at 10 m/s. We consider the number of control actions is $|\mathcal{A}| = 30$, *i.e.*, the allowable heading changes are $\{0, \pi/15, \ldots, 29\pi/15\}$ (rad).

For tracking and planning algorithm *with the terrain awareness*, since the elevation data (z-axis) are already available, we only need to estimate the object position in two dimensions of the xy-axes, then deriving the elevation in z-axis from the DEM data based on its x and y estimated positions. For parameter settings, we implement the same settings as in the case *without terrain awareness*, except the particles of the initial distribution are only sampled from $\mathcal{U}[0, 1000] \times \mathcal{U}[0, 1000]$ for xy-axes, while z-axis particles are calculated from DEM data based on the particles of the xy-axes. Further, given known terrain information, the co-variance matrix of process noise is $Q^{(x)} = [1, 1, 0]^T I_3$ $(\text{m/s})^2$.

Figure B.3 and Figure B.4 depict the tracking and localisation results *with terrain awareness* and *without terrain awareness* algorithms, respectively. Table B.1 provides detailed comparisons between these two approaches over 10 Monte Carlo trials in SITL emulated environments. We can see that the tracking error in term of RMS is similar for both algorithms with or without terrain data.

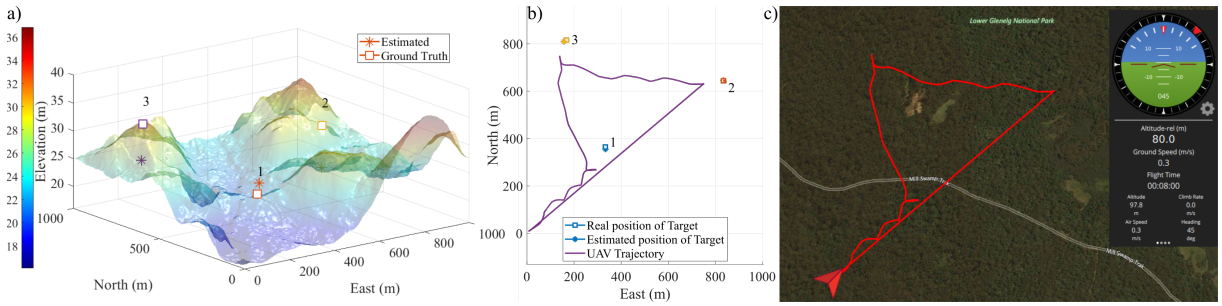


**Figure B.3. The tracking and localisation results without terrain awareness to track and localise three radio-tagged objects in the Lower Glenelg National Park - SA.** a) the ground truth vs the estimated positions in three dimensions (North-East-Elevation); b) the UAV trajectory using the Shannon entropy and its estimated locations in two dimensions (North-East); c) the screen-shot of the QGroundControl with the UAV trajectory.

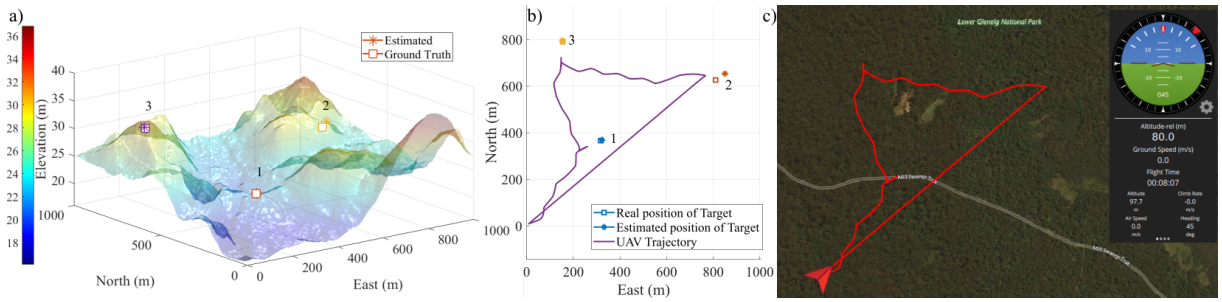


**Figure B.4. The tracking and localisation results with terrain awareness to track and localise three radio-tagged objects in the Lower Glenelg National Park, SA.** a) the ground truth vs the estimated positions in three dimensions (North-East-Elevation); b) the UAV trajectory using the Shannon entropy based reward function and the estimated locations of the radio tags in two dimensions (North-East); c) A screen capture of Software In the Loop simulation with QGroundControl showing the UAV trajectory. Here the straight-line path shows the UAV returning to its home location after the tracking task is complete.

It is expected that the algorithm using terrain information has a smaller z-axis error, which is due to the errors in estimating positions in xy-axes. Further, we notice that the flight time for terrain awareness is significantly shorter because it only needs to estimate two unknown variables compared to the algorithm *without terrain awareness*. Thus, when the terrain information is readily available, we should implement the tracking algorithm with *terrain awareness* to improve flight times. However, most areas in Australia still do not have a Digital Elevation Model, thus implementing our algorithm—tracking *without terrain awareness*—-can play an important role in tracking wildlife objects in unknown terrains. Notably, flight times of approximately,

**Table B.1. Tracking and localising performance over 10 Monte Carlo runs for tracking radio-tagged objects.** Location: the Lower Glenelg National Park, SA

| Terrain Aware | Error (m) | | | RMS (m) | Flight Time (s) |
|---|---|---|---|---|---|
| | x-axis | y-axis | z-axis | | |
| No | 12.6 | 13.4 | 4.2 | 20.78 | 414.4 |
| Yes | 14.4 | 13.5 | 0.3 | 21.53 | **379.1** |

400 seconds for environments without terrain information are easily achievable with modern small size battery powered UAVs.

### B.4.4 Scenario 2:

The second scenario considers the problem for tracking and localising three mobile radio-tagged objects in Dorrigo National Park, New South Wales (NSW). This terrain is more challenging than *Scenario 1* since the Dorrigo National Park site has larger elevation variations ranging from 51.7 m to 318.7 m. Its initial positions in latitude, longitude, and elevation is $[-30.3730, 152.8622, 119.1]^T$, which is converted to $[0, 0, 119.1]^T$ m in the xyz-axes.

For parameters, we apply the same settings as in the *Scenario 1* for the algorithm *with terrain awareness*. For the algorithm *without terrain awareness*, all settings are kept as the same as in the *Scenario 1*, except for the elevation settings. The initial particles for the elevation are sampled from $\mathcal{U}[49.1, 319.1]$ m. Since the variation in the elevation in this site is higher, we set the co-variance matrix of the process noise as $Q^{(x)} = [1, 1, 1]^T$ $(\text{m/s})^2$. Further, the UAV is armed, taken off and changed to an altitude of 400 m AGL[16] using the QGroundControl, *i.e.*, its initial state is set at $u_0 = [10\text{ m}, 10\text{ m}, 519.1\text{ m}, \pi/4\text{ rad}]^T$.

Figure B.5 and Figure B.6 present the tracking and localisation results *with terrain awareness* and *without terrain awareness* algorithms, respectively, for tracking radio-tagged wildlife the Dorrigo National Park, NSW. Here, the elevations change significantly. We can see that our algorithm can still perform well and accurately

[16] We understand that it is legally not possible to fly a UAV at an altitude higher than 120 m AGL (Civil Aviation Safety Authority, 2017). However, as a proof of concept and in an emulated environment, we set the relative altitude to 400 m AGL to remove the obstacle avoidance problem from our formulation. We leave this for future work.

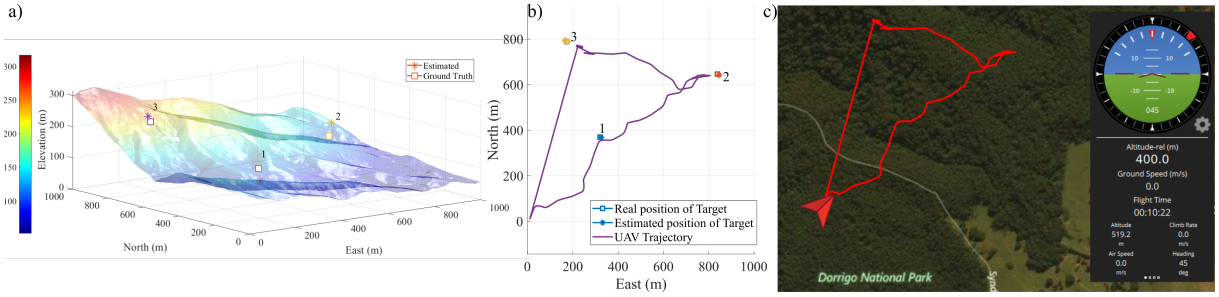


**Figure B.5. The tracking and localisation results without terrain awareness to track and localise three mobile radio-tagged objects in the Dorrigo National Park, NSW.** a) the ground truth vs the estimated positions in three dimensions (North-East-Elevation); b) the UAV trajectory using the Shannon entropy based reward function and the estimated locations of the radio tags in two dimensions (North-East); c) the screen capture of the Software In the Loop simulation with QGroundControl showing the UAV trajectory. Here, the straight line trajectory shows the UAV returning home after completing the tracking task.

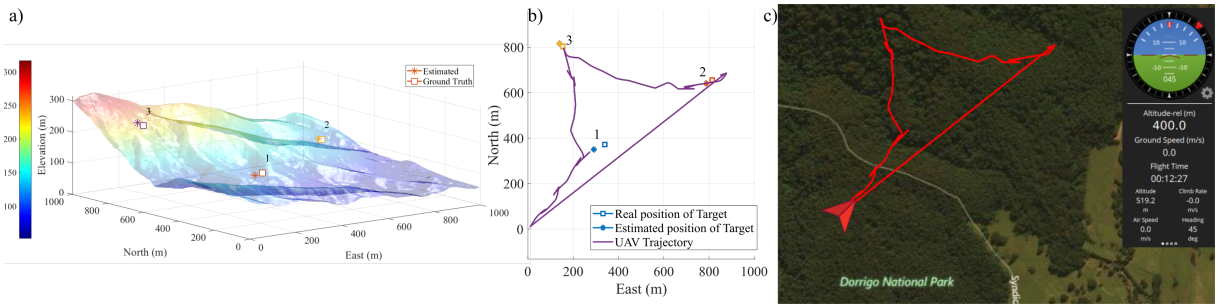


**Figure B.6. The tracking and localisation results with terrain awareness to track and localise three radio-tagged objects in the Dorrigo National Park, NSW.** a) the ground truth vs the estimated positions in three dimensions (North-East-Elevation); b) the UAV trajectory using the Shannon entropy and its estimated locations in two dimensions (North-East); c) the screen capture of the Software In The Loop simulation with QGroundControl showing the UAV trajectory. Again, the straight line path shows the UAV returning to its home location after completing the task.

localise three mobile radio-tagged objects in this challenging survey area. In this particular mission, the RMS and flight time are $(31.8\text{ m}, 705.1\text{ s})$ and $(28.2\text{ m}, 603.3\text{ s})$ for algorithms *without terrain awareness* and *with terrain awareness*, respectively. Although the RMS values are higher compared with those in Table B.1 due to the challenging environment, the results demonstrate the robustness of our proposed algorithm. Our RSSI based measurements based planning for tracking can localise the mobile radio-tagged objects under very challenging terrain variations. Notably, the flight times are longer than with *Scenario 1*, however, flight times of approximately 700 seconds are still achievable with modern battery powered medium size UAVs in

the 2 kg to 4 kg range. For instance, our **TrackerBots** demonstrated in Chapter 3 has a flight time of approximately 6-10 minutes whilst carrying a sensor system payload of mass 260 g.

## B.5 Conclusion

We have validated our formulation for planning to track multiple mobile VHF radio tagged objects in emulated 3D environments using a measurement model validated in field experiments using a software in the loop simulations. Therefore, we have taken the first steps towards three dimensional tracking and planning for a UAV using RSSI-based method with or without terrain awareness. Whilst we have demonstrated that it is possible to conduct 3D tracking, the particular limitation of the study is that the signals generated in the study have only considered the unencumbered propagation of the light-of-sight signal transmitted from a radio beacon and have not considered the signal propagation complexities of a 3D environment and we did not consider the problems of missed detections and false detections or the limited FoV of the onboard sensor.

# Bibliography


Amato, C. and Oliehoek, F., 2015. Scalable planning and learning for multiagent POMDPs. *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. pp.1995–2002. (Cited on pages 18 and 93.)

Anderson, C., 2017. Drones go to work. *Harvard Business Review* [Online]. Available from: https://hbr.org/cover-story/2017/05/drones-go-to-work/ [Accessed 20 Mar 2018]. (Cited on page 2.)

Araya, M., Buffet, O., Thomas, V. and Charpillet, F., 2010. A POMDP extension with belief-dependent rewards. *Proceedings of the 23rd International Conference on Neural Information Processing Systems*. pp.64–72. (Cited on page 93.)

Artieda, J., Sebastian, J.M., Campoy, P., Correa, J.F., Mondragón, I.F., Martínez, C. and Olivares, M., 2009. Visual 3-D SLAM from UAVs. *Journal of Intelligent and Robotic Systems*, 55(4-5), pp.299 – 321. (Cited on page 152.)

Arulampalam, M.S., Maskell, S., Gordon, N. and Clapp, T., 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), pp.174–188. (Cited on page 53.)

Australia-Geoscience, 2018. ELVIS - elevation and depth - foundation spatial data [Online]. Available from: http://elevation.fsdf.org.au/ [Accessed 20 Sep 2018]. (Cited on pages 158, 162, and 164.)

Baek, S.S., Kwon, H., Yoder, J.A. and Pack, D., 2013. Optimal path planning of a target-following fixed-wing UAV using sequential decision processes. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp.2955–2962. (Cited on pages 18 and 63.)

Bar-Shalom, Y., 1987. *Tracking and data association*. Academic Press Professional, Inc. (Cited on pages 12, 15, and 26.)

Barniv, Y., 1985. Dynamic programming solution for detecting dim moving targets. *IEEE Transactions on Aerospace and Electronic Systems*, AES-21(1), pp.144–156. (Cited on pages 64 and 74.)

Battistelli, G. and Chisci, L., 2014. Kullback-Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. *Automatica*, 50(3), pp.707 – 718. (Cited on pages 121 and 138.)

Battistelli, G., Chisci, L., Fantacci, C., Farina, A. and Graziano, A., 2013. Consensus CPHD filter for distributed multitarget tracking. *IEEE Journal of Selected Topics in Signal Processing*, 7(3), pp.508–520. (Cited on pages 3, 4, 120, and 121.)

Beard, M., Vo, B.T. and Vo, B., 2020. A Solution for Large-Scale Multi-Object Tracking. *IEEE Transactions on Signal Processing*, pp.1–16. (Cited on pages 23, 24, and 126.)

Beard, M., Vo, B.T. and Vo, B.N., 2017. OSPA(2): using the OSPA metric to evaluate multi-target tracking performance. *Proceedings of the International Conference on Control, Automation and Information Sciences*. pp.86–91. (Cited on pages 23, 122, and 139.)

Beard, M., Vo, B.T. and Vo, B.N., 2018. Performance evaluation for large-scale multi-target tracking algorithms. *Proceedings of the 21st International Conference on Information Fusion*. pp.1–5. (Cited on pages 23 and 122.)

Beard, M., Vo, B.T., Vo, B.N. and Arulampalam, S., 2015. Sensor control for multi-target tracking using Cauchy-Schwarz divergence. *Proceedings of the 18th International Conference on Information Fusion*. pp.937–944. (Cited on page 20.)

Beard, M.A., Vo, B.T., Vo, B.N. and Arulampalam, S., 2017. Void probabilities and Cauchy-Schwarz divergence for generalized labeled multi-Bernoulli models. *IEEE Transactions on Signal Processing*, 65. (Cited on pages 20, 33, 57, 63, 64, 77, 79, 80, 94, 95, and 161.)

Beck, Z., Teacy, W.L., Rogers, A. and Jennings, N.R., 2018. Collaborative online planning for automated victim search in disaster response. *Robotics and Autonomous Systems*, 100, pp.251–266. (Cited on pages 3, 92, and 96.)

Bernstein, D.S., Givan, R., Immerman, N. and Zilberstein, S., 2002. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4), pp.819–840. (Cited on pages 18, 93, 118, and 151.)

Bertsekas, D.P. et al., 1996. *Dynamic programming and optimal control*, vol. 1. Athena Scientific Belmont, Massachusetts. (Cited on page 19.)

Blackman, S. and Popoli, R., 1999. *Design and analysis of modern tracking systems*. Artech House. (Cited on pages 3, 15, and 94.)

Buzzi, S., Lops, M. and Venturino, L., 2005. Track-before-detect procedures for early detection of moving target from airborne radars. *IEEE Transactions on Aerospace and Electronic Systems*, 41(3), pp.937–954. (Cited on pages 64 and 75.)

Buzzi, S., Lops, M., Venturino, L. and Ferri, M., 2008. Track-before-detect procedures in a multi-target environment. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3), pp.1135–1150. (Cited on pages 64 and 75.)

Caballero, F., Merino, L., Maza, I. and Ollero, A., 2008. A particle filtering method for wireless sensor network localization with an aerial robot beacon. *Proceedings of the IEEE International Conference on Robotics and Automation*. pp.596–601. (Cited on page 28.)

Calafiore, G.C. and Abrate, F., 2009. Distributed linear estimation over sensor networks. *International Journal of Control*, 82(5), pp.868–882. (Cited on page 138.)

Castañón, D.A. and Carin, L., 2008. Stochastic control theory for sensor management. *Foundations and Applications of Sensor Management*. Springer, pp.7–32. (Cited on page 77.)

Cattivelli, F.S. and Sayed, A.H., 2010. Diffusion strategies for distributed kalman filtering and smoothing. *IEEE Transactions on Automatic Control*, 55(9), pp.2069–2084. (Cited on page 138.)

Chang, K.C., Chong, C.Y. and Mori, S., 2010. Analytical and computational evaluation of scalable distributed fusion algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 46(4), pp.2022–2034. (Cited on page 121.)

Chang, K.C., Saha, R.K. and Bar-Shalom, Y., 1997. On optimal track-to-track fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 33(4), pp.1271–1276. (Cited on page 122.)

Charrow, B., Michael, N. and Kumar, V., 2015. Active control strategies for discovering and localizing devices with range-only sensors. *Algorithmic Foundations of Robotics XI*. Springer, pp.55–71. (Cited on pages 58, 94, 95, 96, 100, 159, 160, and 162.)

Chong, C.Y., Mori, S., Barker, W.H. and Chang, K.C., 2000. Architectures and algorithms for track association and fusion. *IEEE Aerospace and Electronic Systems Magazine*, 15(1), pp.5–13. (Cited on pages 122 and 133.)

Chong, C.Y., Mori, S. and Chang, K.C., 1990. Distributed multitarget multisensor tracking. *Multitarget-multisensor tracking: Advanced applications*, pp.247–296. (Cited on page 122.)

Christiansen, P., Steen, K.A., Jørgensen, R.N. and Karstoft, H., 2014. Automated detection and recognition of wildlife using thermal cameras. *Sensors*, 14(8), pp.13778–13793. (Cited on page 26.)

Chung, S., Paranjape, A.A., Dames, P., Shen, S. and Kumar, V., 2018. A survey on aerial swarm robotics. *IEEE Transactions on Robotics*, 34(4), pp.837–855. (Cited on page 2.)

Civil Aviation Authority Of New Zealand, 2015. Advisory circular AC101-1 [Online]. Available from: `https://www.caa.govt.nz/assets/legacy/Advisory_Circulars/AC101-1.pdf` [Accessed 1 Sep 2018]. (Cited on page 55.)

Civil Aviation Safety Authority, 2017. AC 101-10 remotely piloted aircraft systems - operation of excluded RPA (other than model aircraft) [Online]. Available from: `https://www.casa.gov.au/files/ac10110pdf` [Accessed 13 Apr 2018]. (Cited on pages 53, 55, and 167.)

Clark, D., Julier, S., Mahler, R. and Ristic, B., 2010. Robust multi-object sensor fusion with unknown correlations. *Proceedings of the Sensor Signal Processing for Defence*. pp.1–5. (Cited on page 121.)

Cliff, O.M., Fitch, R., Sukkarieh, S., Saunders, D. and Heinsohn, R., 2015. Online localization of radio-tagged wildlife with an autonomous aerial robot system. *Proceedings of the Robotics: Science and Systems*. (Cited on pages xxv, 2, 28, 29, 33, 42, 43, 44, 50, 52, 53, 54, 55, 58, 62, 65, 92, 96, 106, 160, and 162.)

Cochran, W.W. and Lord Jr, R.D., 1963. A radio-tracking system for wild animals. *The Journal of Wildlife Management*, pp.9–24. (Cited on page 26.)

Coello, C.A.C., Lamont, G.B., Van Veldhuizen, D.A. et al., 2007. *Evolutionary algorithms for solving multi-objective problems*, vol. 5. Springer. (Cited on page 102.)

Cover, T.M. and Thomas, J.A., 2012. *Elements of information theory*. John Wiley & Sons. (Cited on pages 98, 99, and 121.)

Cox, I.J. and Hingorani, S.L., 1996. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2), pp.138–150. (Cited on page 3.)

Daley, D.J. and Vere-Jones, D., 2007. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media. (Cited on page 63.)

Dames, P., Tokekar, P. and Kumar, V., 2017. Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots. *The International Journal of Robotics Research*, 36(13-14), pp.1540–1553. (Cited on pages 18, 93, 94, and 95.)

Dames, P.M. and Kumar, V., 2015. Autonomous localization of an unknown number of targets without data association using teams of mobile sensors. *IEEE Transactions on Automation Science and Engineering*, 12(3), pp.850–864. (Cited on pages 18, 93, and 94.)

Davenport, W. and Root, W., 1987. *An introduction to the theory of random signals and noise*. Wiley-IEEE Press. (Cited on page 72.)

Dos Santos, G.A.M., Barnes, Z., Lo, E., Ritoper, B., Nishizaki, L., Tejeda, X., Ke, A., Lin, H., Schurgers, C., Lin, A. et al., 2014. Small unmanned aerial vehicle system for wildlife radio collar tracking. *Proceedings of the 11th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*. pp.761–766. (Cited on pages 28, 54, and 55.)

Douc, R. and Cappé, O., 2005. Comparison of resampling schemes for particle filtering. *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*. IEEE, pp.64–69. (Cited on page 14.)

Doucet, A., De Freitas, N. and Gordon, N., 2001. An introduction to sequential Monte Carlo methods. *Sequential Monte Carlo Methods in Practice*. Springer, pp.3–14. (Cited on pages 13 and 14.)

Dunne, D. and Kirubarajan, T., 2013. Multiple model multi-Bernoulli filters for manoeuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, 49(4), pp.2679–2692. (Cited on pages 64 and 75.)

Ebenezer, S.P. and Papandreou-Suppappola, A., 2016. Generalized recursive track-before-detect with proposal partitioning for tracking varying number of

multiple targets in low SNR. *IEEE Transactions on Signal Processing*, 64(11), pp.2819–2834. (Cited on page 64.)

Elfes, A., 1989. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6), pp.46–57. (Cited on page 100.)

Fantacci, C., Vo, B.N., Vo, B.T., Battistelli, G. and Chisci, L., 2015. Consensus labeled random finite set filtering for distributed multi-object tracking. *arXiv preprint arXiv:1501.01579*. (Cited on page 121.)

Fantacci, C., Vo, B.N., Vo, B.T., Battistelli, G. and Chisci, L., 2018. Robust fusion for multisensor multiobject tracking. *IEEE Signal Processing Letters*, 25(5), pp.640–644. (Cited on pages 4, 121, and 138.)

Federal Ministry of Transport and Digital Infrastructure, 2017. New rules governing drones in force [Online]. Available from: https://www.bmvi.de/SharedDocs/EN/PressRelease/2017/050-new-rules-governing-drones.html [Accessed 1 Sep 2018]. (Cited on pages 53 and 55.)

Finnish Transport Safety Agency, 2016. Use of remotely piloted aircraft and model aircraft [Online]. Available from: https://www.trafi.fi/filebank/a/1483970125/4a6ac53bf4b1cb434d7f85a15f36dde0/23661-OPS_M1-32_RPAS_2016_eng.pdf [Accessed 1 Sep 2018]. (Cited on page 55.)

Fossel, J., Hennes, D., Claes, D., Alers, S. and Tuyls, K., 2013. OctoSLAM: A 3D mapping approach to situational awareness of unmanned aerial vehicles. *Proceedings of the International Conference on Unmanned Aircraft Systems*. pp.179–188. (Cited on page 152.)

Gao, L., Battistelli, G. and Chisci, L., 2019. Event-triggered distributed multitarget tracking. *IEEE Transactions on Signal and Information Processing over Networks*, 5(3), pp.570–584. (Cited on page 121.)

Gerasenko, S., Joshi, A., Rayaprolu, S., Ponnavaikko, K. and Agrawal, D.P., 2001. Beacon signals: what, why, how, and where? *Computer*, 34(10), pp.108–110. (Cited on pages 2, 62, 65, and 92.)

Gonzalez, L.F., Montes, G.A., Puig, E., Johnson, S., Mengersen, K. and Gaston, K.J., 2016. Unmanned aerial vehicles (UAVs) and artificial intelligence revolutionizing wildlife monitoring and conservation. *Sensors*, 16(1), pp.97 – 114. (Cited on page 26.)

Gordon, N., 1997. A hybrid bootstrap filter for target tracking in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1), pp.353–358. (Cited on page 13.)

Gordon, N.J., Salmond, D.J. and Smith, A.F., 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F - Radar and Signal Processing*, 140(2), pp.107–113. (Cited on pages 13 and 30.)

Gostar, A., Hoseinnezhad, R. and Bab-hadiashar, A., 2016. Multi-Bernoulli sensor control using Cauchy-Schwarz divergence. *Proceedings of the 19th International Conference on Information Fusion*. pp.651–657. (Cited on pages 77 and 78.)

Gostar, A.K., Hoseinnezhad, R. and Bab-Hadiashar, A., 2013. Robust multi-Bernoulli sensor selection for multi-target tracking in sensor networks. *IEEE Signal Processing Letters*, 20(12), pp.1167–1170. (Cited on pages 20 and 63.)

Gostar, A.K., Hoseinnezhad, R. and Bab-Hadiashar, A., 2016. Multi-Bernoulli sensor-selection for multi-target tracking with unknown clutter and detection profiles. *Signal Processing*, 119, pp.28 – 42. (Cited on page 94.)

Gostar, A.K., Hoseinnezhad, R., Bab-Hadiashar, A. and Liu, W., 2017. Sensor-management for multitarget filters via minimization of posterior dispersion. *IEEE Transactions on Aerospace and Electronic Systems*, 53(6), pp.2877–2884. (Cited on pages 20, 63, and 77.)

Harris, F., 1978. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1), pp.51–83. (Cited on page 74.)

Hero, A.O., Kreucher, C.M. and Blatt, D., 2008. *Information theoretic approaches to sensor management*. Springer US. (Cited on page 33.)

Hoang, H.G., Vo, B.N., Vo, B.T. and Mahler, R., 2015. The Cauchy–Schwarz divergence for Poisson point processes. *IEEE Transactions on Information Theory*, 61(8), pp.4475–4485. (Cited on pages 21 and 63.)

Hoang, H.G. and Vo, B.T., 2014. Sensor management for multi-target tracking via multi-Bernoulli filtering. *Automatica*, 50(4), pp.1135–1142. (Cited on pages 63, 77, 78, and 94.)

Hodgson, J.C. and Koh, L.P., 2016. Best practice for minimising unmanned aerial vehicle disturbance to wildlife in biological field research. *Current Biology*, 26(10), pp.R404–R405. (Cited on page 57.)

Hoffman, J.R. and Mahler, R.P.S., 2004. Multitarget miss distance via optimal assignment. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 34(3), pp.327–336. (Cited on page 124.)

Hol, J.D., Schon, T.B. and Gustafsson, F., 2006. On resampling algorithms for particle filters. *Proceedings of the IEEE Nonlinear Statistical Signal Processing Workshop*. pp.79–82. (Cited on page 14.)

Hoseinnezhad, R., Vo, B.N., Vo, B.T. and Suter, D., 2012. Visual tracking of numerous targets via multi-Bernoulli filtering of image data. *Pattern Recognition*, 45(10), pp.3625 – 3635. (Cited on page 3.)

Hsu, D., Lee, W.S. and Rong, N., 2008. A point-based POMDP planner for target tracking. *Proceedings of the IEEE International Conference on Robotics and Automation*. pp.2644–2650. (Cited on pages 19 and 33.)

Hurley, M.B., 2002. An information theoretic justification for covariance intersection and its generalization. *Proceedings of the 5th International Conference on Information Fusion*. vol. 1, pp.505–511. (Cited on page 121.)

Jakes, W.C., 1974. *Microwave mobile communications*. Wiley, New York. (Cited on page 31.)

Jazwinski, A.H., 2007. *Stochastic processes and filtering theory*. Courier Corporation. (Cited on page 12.)

Jensen, A.M., Geller, D.K. and Chen, Y., 2014. Monte Carlo simulation analysis of tagged fish radio tracking performance by swarming unmanned aerial vehicles in fractional order potential fields. *Journal of Intelligent & Robotic Systems*, 74(1-2), pp.287–307. (Cited on page 28.)

Julier, S., Uhlmann, J. and Durrant-Whyte, H.F., 2000. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3), pp.477–482. (Cited on page 12.)

Julier, S.J., Bailey, T. and Uhlmann, J.K., 2006. Using exponential mixture models for suboptimal distributed data fusion. *Proceedings of the IEEE Nonlinear Statistical Signal Processing Workshop*. pp.160–163. (Cited on page 121.)

Julier, S.J. and Uhlmann, J.K., 2004. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3), pp.401–422. (Cited on page 12.)

Kaelbling, L.P., Littman, M.L. and Cassandra, A.R., 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1), pp.99–134. (Cited on pages 18, 32, and 77.)

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), pp.35–45. (Cited on page 11.)

Kaplan, L.M., Bar-Shalom, Y. and Blair, W.D., 2008. Assignment costs for multiple sensor track-to-track association. *IEEE Transactions on Aerospace and Electronic Systems*, 44(2), pp.655–677. (Cited on page 122.)

Kays, R., Tilak, S., Crofoot, M., Fountain, T., Obando, D., Ortega, A., Kuemmeth, F., Mandel, J., Swenson, G., Lambert, T. et al., 2011. Tracking animal location and activity with an automated radio telemetry system in a tropical rainforest. *The Computer Journal*, pp.1931–1948. (Cited on pages 2, 26, 28, 62, 65, 92, and 96.)

Kendall, W.S., Mecke, J. and Stoyan, D., 1995. *Stochastic geometry and its applications*. John Wiley & Sons. (Cited on page 80.)

Kenward, R.E., 2000. *A manual for wildlife radio tagging*. Academic Press. (Cited on pages 26 and 65.)

Körner, F., Speck, R., Göktogan, A.H. and Sukkarieh, S., 2010. Autonomous airborne wildlife tracking using radio signal strength. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp.107–112. (Cited on page 28.)

Koski, J., 1993. *Multicriterion structural optimization*, Dordrecht: Springer Netherlands, pp.793–809. (Cited on page 102.)

Krause, A., McMahan, H.B., Guestrin, C. and Gupta, A., 2008. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(Dec), pp.2761–2801. (Cited on pages 102, 110, and 111.)

Krause, A., Singh, A. and Guestrin, C., 2008. Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb), pp.235–284. (Cited on page 99.)

Kreucher, C., Hero, A.O. and Kastella, K., 2005. A comparison of task driven and information driven sensor management for target tracking. *Proceedings of the 44th IEEE Conference on Decision and Control*. pp.4004–4009. (Cited on page 20.)

Kuhn, H.W., 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2), pp.83–97. (Cited on page 129.)

Lange, S., Sunderhauf, N. and Protzel, P., 2009. A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments. *Proceedings of the International Conference on Advanced Robotics*. pp.1–6. (Cited on page 152.)

Lehmann, F., 2012. Recursive Bayesian filtering for multitarget track-before-detect in passive radars. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3), pp.2458–2480. (Cited on pages 62, 64, and 75.)

Levina, E. and Bickel, P., 2001. The Earth Mover's distance is the Mallows distance: some insights from statistics. *Proceedings of the 8th IEEE International Conference on Computer Vision*. vol. 2, pp.251–256 vol.2. (Cited on page 124.)

Li, G., Battistelli, G., Yi, W. and Kong, L., 2020. Distributed multi-sensor multi-view fusion based on generalized covariance intersection. *Signal Processing*, 166, pp.107246 – 107258. (Cited on pages 121 and 140.)

Li, S., Battistelli, G., Chisci, L., Yi, W., Wang, B. and Kong, L., 2019. Computationally efficient multi-agent multi-object tracking with labeled random finite sets. *IEEE Transactions on Signal Processing*, 67(1), pp.260–275. (Cited on pages 121, 122, 127, 129, and 140.)

Li, S., Yi, W., Hoseinnezhad, R., Battistelli, G., Wang, B. and Kong, L., 2018. Robust distributed fusion with labeled random finite sets. *IEEE Transactions on Signal Processing*, 66(2), pp.278–293. (Cited on pages 4, 121, and 122.)

Liggins, M.E., Chong, C.Y., Kadar, I., Alford, M.G., Vannicola, V. and Thomopoulos, S., 1997. Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, 85(1), pp.95–107. (Cited on page 120.)

Liu, H., Liu, M., Wei, X., Song, Q., Ge, Y. and Wang, F., 2014. Auto altitude holding of quadrotor UAVs with Kalman filter based vertical velocity estimation. *Proceedings of the 11th World Congress on Intelligent Control and Automation*. pp.4765–4770. (Cited on page 150.)

Lovejoy, William S, 1991. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research*, 28(1), pp.47–65. (Cited on page 19.)

Luo, Z.Q., Gastpar, M., Liu, J. and Swami, A., 2006. Distributed signal processing in sensor networks [from the guest editors]. *IEEE Signal Processing Magazine*, 23(4), pp.14–15. (Cited on pages 3 and 120.)

MacDermed, L.C. and Isbell, C.L., 2013. Point based value iteration with optimal belief compression for Dec-POMDPs. *Proceedings of the 26th International Conference on Neural Information Processing Systems*. pp.100–108. (Cited on page 93.)

Mahler, R., 2004. Multitarget sensor management of dispersed mobile sensors. *Theory and Algorithms for Cooperative Systems*. World Scientific, pp.239–310. (Cited on page 97.)

Mahler, R., 2007a. PHD filters of higher order in target number. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4). (Cited on pages 18 and 121.)

Mahler, R.P., 2000. Optimal/robust distributed data fusion: a unified approach. *Signal Processing, Sensor Fusion, and Target Recognition IX*. International Society for Optics and Photonics, vol. 4052, pp.128–138. (Cited on pages 3, 4, 120, and 121.)

Mahler, R.P., 2003. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic systems*, 39(4), pp.1152–1178. (Cited on pages 18 and 121.)

Mahler, R.P., 2007b. *Statistical multisource-multitarget information fusion*. Artech House. (Cited on pages 3, 15, 16, 17, 18, 63, 94, 96, 98, and 121.)

Mahler, R.P., 2014. *Advances in statistical multisource-multitarget information fusion*. Artech House. (Cited on page 3.)

Messias, J.V., Spaan, M. and Lima, P.U., 2011. Efficient offline communication policies for factored multiagent POMDPs. *Proceedings of the 24th International Conference on Neural Information Processing Systems*. pp.1917–1925. (Cited on pages 18, 93, and 151.)

Miller, M.L., Stone, H.S. and Cox, I.J., 1997. Optimizing Murty's ranked assignment method. *IEEE Transactions on Aerospace and Electronic Systems*, 33(3), pp.851–862. (Cited on page 129.)

Moller, J. and Waagepetersen, R.P., 2003. *Statistical inference and simulation for spatial point processes*. Chapman and Hall/CRC. (Cited on page 63.)

Monahan, G.E., 1982. State of the art — survey of partially observable Markov decision processes: theory, models, and algorithms. *Management Science*, 28(1), pp.1–16. (Cited on page 19.)

Mori, S., Barker, W.H., Chong, C.Y. and Chang, K.C., 2002. Track association and track fusion with nondeterministic target dynamics. *IEEE Transactions on Aerospace and Electronic Systems* [Online], 38(2), pp.659–668. Available from: https://doi.org/10.1109/TAES.2002.1008994. (Cited on page 122.)

Mori, S., Chang, K. and Chong, C., 2014. Performance prediction of feature-aided track-to-track association. *IEEE Transactions on Aerospace and Electronic Systems*, 50(4), pp.2593–2603. (Cited on page 122.)

Mori, S. and Chong, C.Y., 2003. Track-to-track association metric I.I.D.-non-poisson cases. *Proceedings of the 6th International Conference of Information Fusion*. vol. 1, pp.553–559. (Cited on page 122.)

Mulero-Pázmány, M., Jenni-Eiermann, S., Strebel, N., Sattler, T., Negro, J.J. and Tablado, Z., 2017. Unmanned aircraft systems as a new source of disturbance for wildlife: a systematic review. *PLOS ONE*, 12(6), pp.1–14. (Cited on page 57.)

Mullane, J., Vo, B., Adams, M.D. and Vo, B., 2011. A random-finite-set approach to Bayesian SLAM. *IEEE Transactions on Robotics*, 27(2), pp.268–282. (Cited on page 3.)

Munkres, J., 1957. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1), pp.32–38. (Cited on page 129.)

Munz, M., Mählisch, M. and Dietmayer, K., 2010. Generic centralized multi sensor data fusion based on probabilistic sensor and environment models for driver assistance systems. *IEEE Intelligent Transportation Systems Magazine*, 2(1), pp.6–17. (Cited on page 3.)

Murphy, R.R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H. and Erkmen, A.M., 2008. *Search and rescue robotics*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp.1151–1173. (Cited on pages 2, 62, 65, and 92.)

Murty, K.G., 1968. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16(3), pp.682–687. (Cited on page 129.)

Nair, R., Varakantham, P., Tambe, M. and Yokoo, M., 2005. Networked distributed POMDPs: a synthesis of distributed constraint optimization and POMDPs. *Proceedings of the 20th AAAI Conference on Artificial Intelligence*. vol. 5, pp.133–139. (Cited on pages 18, 93, and 151.)

Nemhauser, G.L., Wolsey, L.A. and Fisher, M.L., 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1), pp.265–294. (Cited on pages 100, 101, 102, and 103.)

Nguyen, H.V., Chesser, M., Chen, F., Rezatofighi, S.H. and Ranasinghe, D.C., 2018a. Demo abstract: autonomous UAV sensor system for searching and locating VHF radio-tagged wildlife. *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. ACM, pp.333–334. (Cited on page 160.)

Nguyen, H.V., Chesser, M., Koh, L.P., Rezatofighi, H. and Ranasinghe, D.C., 2019a. TrackerBots: autonomous unmanned aerial vehicle for real-time localization and tracking of multiple radio-tagged animals. *Journal of Field Robotics*, 36(3), pp.617–635. (Cited on pages 5, 62, 63, 65, 66, 67, 82, 92, 96, 159, 160, and 161.)

Nguyen, H.V., Rezatofighi, H., Taggart, D., Ostendorf, B. and Ranasinghe, D.C., 2018b. TrackerBots: software in the loop study of quad-copter robots for locating radio-tags in a 3D space. *Proceedings of the Australasian Conference on Robotics and Automation*. Lincoln, New Zealand, pp.304–313. (Cited on page 96.)

Nguyen, H.V., Rezatofighi, H., Vo, B.N. and Ranasinghe, D.C., 2019b. Online UAV path planning for joint detection and tracking of multiple radio-tagged objects. *IEEE Transactions on Signal Processing*, 67(20), pp.5365–5379. (Cited on pages 6 and 96.)

Nguyen, H.V., Rezatofighi, H., Vo, B.N. and Ranasinghe, D.C., 2020a. Distributed multi-sensor multi-object tracking under limited field of view sensors. *Prepared to submit to IEEE Transactions on Signal Processing*. (Cited on page 6.)

Nguyen, H.V., Rezatofighi, H., Vo, B.N. and Ranasinghe, D.C., 2020b. Multi-objective multi-agent planning for jointly discovering and tracking mobile objects. *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. (Cited on page 6.)

Office Of The Director General Of Civil Aviation, 2018. Requirements for operation of civil remotely piloted aircraft system (RPAS) [Online]. Available from: `http://dgca.nic.in/cars/D3X-X1.pdf` [Accessed 1 Sep 2018]. (Cited on pages 53 and 55.)

Olfati-Saber, R., 2007. Distributed Kalman filtering for sensor networks. *Proceedings of the 46th IEEE Conference on Decision and Control*. pp.5492–5498. (Cited on page 138.)

Oliehoek, F.A., Spaan, M.T., Whiteson, S. and Vlassis, N., 2008. Exploiting locality of interaction in factored Dec-POMDPs. *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*. pp.517–524. (Cited on page 151.)

Olivares-Mendez, M.A., Fu, C., Ludivig, P., Bissyandé, T.F., Kannan, S., Zurad, M., Annaiyan, A., Voos, H. and Campoy, P., 2015. Towards an autonomous vision-based unmanned aerial system against wildlife poachers. *Sensors*, 15(12), pp.31362–31391. (Cited on page 26.)

Orfanidis, S.J., 2002. *Electromagnetic waves and antennas*. Rutgers University New Brunswick, NJ. (Cited on pages 44 and 66.)

Ossmann, M., 2015. Software defined radio with HackRF [Online]. Available from: `https://greatscottgadgets.com/sdr/` [Accessed 19 Jan 2017]. (Cited on pages 36 and 66.)

Papi, F., Vo, B.N., Vo, B.T., Fantacci, C. and Beard, M., 2015. Generalized labeled multi-Bernoulli approximation of multi-object densities. *IEEE Transactions on Signal Processing*, 63(20), pp.5487–5497. (Cited on pages 64 and 75.)

Papi, F., Vo, B.T., Bocquel, M. and Vo, B.N., 2013. Multi-target track-before-detect using labeled random finite set. *Proceedings of the International Conference on Control, Automation and Information Science*. pp.116–121. (Cited on page 75.)

Patwari, N., Ash, J.N., Kyperountas, S., Hero, A.O., Moses, R.L. and Correal, N.S., 2005. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4), pp.54–69. (Cited on page 31.)

Pedersen, C.R., Nielsen, L.R. and Andersen, K.A., 2008. An algorithm for ranking assignments using reoptimization. *Computers & Operations Research*, 35(11), pp.3714–3726. (Cited on page 129.)

Posch, A. and Sukkarieh, S., 2009. UAV based search for a radio tagged animal using particle filters. *Proceedings of the Australasian Conference on Robotics and Automation*. pp.2–4. (Cited on page 28.)

Ragi, S. and Chong, E.K.P., 2013. UAV path planning in a dynamic environment via partially observable Markov decision process. *IEEE Transactions on Aerospace and Electronic Systems*, 49(4), pp.2397–2412. (Cited on pages 18 and 63.)

Reid, D., 1979. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6), pp.843–854. (Cited on pages 15 and 94.)

Reuter, S., Danzer, A., Stübler, M., Scheel, A. and Granström, K., 2017. A fast implementation of the labeled multi-Bernoulli filter using Gibbs sampling. *Proceedings of the IEEE Intelligent Vehicles Symposium*. pp.765–772. (Cited on page 3.)

Reuter, S., Scheel, A. and Dietmayer, K., 2015. The multiple model labeled multi-Bernoulli filter. *Proceedings of the 18th International Conference on Information Fusion*. pp.1574–1580. (Cited on page 75.)

Reuter, S., Vo, B.T., Vo, B.N. and Dietmayer, K., 2014. The labeled multi-Bernoulli filter. *IEEE Transactions on Signal Processing*, 62(12), pp.3246–3260. (Cited on pages 16, 18, 76, 89, 121, and 139.)

Richards, M.A., 2013. *The discrete-time Fourier transform and discrete Fourier transform of windowed stationary white noise* [Online]. Georgia Institute of Technology. Available from: `http://pwp.gatech.edu/wp-content/uploads/sites/462/2016/08/DFT-of-Noise.pdf` [Accessed 5 Jun 2017]. (Cited on pages 72 and 73.)

Ristic, B., 2013. *Particle filters for random set models*. Springer-Verlag New York. (Cited on page 33.)

Ristic, B., Arulampalam, S. and Gordon, N.c., 2004. *Beyond the Kalman filter : particle filters for tracking applications*. Artech House. (Cited on pages 13, 14, and 30.)

Ristic, B., Morelande, M. and Gunatilaka, A., 2010. Information driven search for point sources of gamma radiation. *Signal Processing*, 90(4), pp.1225–1239. (Cited on pages 41 and 42.)

Ristic, B. and Vo, B.N., 2010. Sensor control for multi-object state-space estimation using random finite sets . *Automatica* , 46(11), pp.1812 – 1818. (Cited on pages 20, 33, 41, 63, 77, 94, and 161.)

Ristic, B., Vo, B.N. and Clark, D., 2011. A note on the reward function for PHD filters with sensor control. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2), pp.1521–1529. (Cited on pages 63 and 77.)

Rizk, Y., Awad, M. and Tunstel, E.W., 2018. Decision making in multiagent systems: a survey. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3), pp.514–529. (Cited on pages 18, 93, and 151.)

Roberge, V., Tarbouchi, M. and Labonte, G., 2013. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Transactions on Industrial Informatics*, 9(1), pp.132–141. (Cited on pages 18 and 63.)

Rutten, M.G., Gordon, N.J. and Maskell, S., 2005. Recursive track-before-detect with target amplitude fluctuations. *IEE Proceedings-Radar, Sonar and Navigation*, 152(5), pp.345–352. (Cited on pages 64 and 74.)

Ryan, T., Marker, P., Roche, R. and Willer, H., 2015. Dronekit-SITL: SITL runner for dronekit. [Online]. Available from: `https://github.com/dronekit/dronekit-sitl` [Accessed 24 Mar 2018]. (Cited on page 162.)

Särkkä, S., Viikari, V. and Jaakkola, K., 2014. RFID-based butterfly location sensing system. *Proceedings of the European Signal Processing Conference*. pp.2045–2049. (Cited on page 28.)

Satsangi, Y., Whiteson, S., Oliehoek, F.A. and Spaan, M.T., 2018. Exploiting submodular value functions for scaling up active perception. *Autonomous Robots*, 42(2), pp.209–233. (Cited on page 93.)

Schartel, M., Burr, R., Schoeder, P., Rossi, G., Hügler, P., Mayer, W. and Waldschmidt, C., 2018. Radar-based altitude over ground estimation of UAVs. *Proceedings of the 11th German Microwave Conference*. pp.103–106. (Cited on page 150.)

Schmuck, P. and Chli, M., 2017. Multi-UAV collaborative monocular SLAM. *Proceedings of the IEEE International Conference on Robotics and Automation*. pp.3863–3870. (Cited on page 152.)

Scholkmann, F., Boss, J. and Wolf, M., 2012. An efficient algorithm for automatic peak detection in noisy periodic and quasi-periodic signals. *Algorithms*, 5(4), pp.588–603. (Cited on page 89.)

Schuhmacher, D., Vo, B.T. and Vo, B.N., 2008. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 56(8), pp.3447–3457. (Cited on pages 22, 86, 104, 124, and 139.)

Selby, W., Corke, P. and Rus, D., 2011. Autonomous aerial navigation and tracking of marine animals. *Proceedings of the Australian Conference on Robotics and Automation*. (Cited on page 26.)

Silver, D. and Veness, J., 2010. Monte-Carlo planning in large POMDPs. *Proceedings of the 23rd International Conference on Neural Information Processing Systems*. pp.2164–2172. (Cited on pages 33 and 93.)

Smith III, J.O., 2011. *Spectral audio signal processing*. W3K publishing. (Cited on pages 69 and 70.)

Song, Y., Xian, B., Zhang, Y., Jiang, X. and Zhang, X., 2015. Towards autonomous control of quadrotor unmanned aerial vehicles in a GPS-denied urban area via laser ranger finder. *Optik*, 126(23), pp.3877 – 3882. (Cited on page 152.)

Souza, E.L., Nakamura, E.F. and Pazzi, R.W., 2016. Target tracking for sensor networks: a survey. *ACM Computing Surveys*, 49(2). (Cited on page 120.)

Spaan, M.T., Veiga, T.S. and Lima, P.U., 2015. Decision-theoretic planning under uncertainty with information rewards for active cooperative perception. *Autonomous Agents and Multi-Agent Systems*, pp.1157–1185. (Cited on page 93.)

Stanković, S.S., Stanković, M.S. and Stipanović, D.M., 2009. Consensus based overlapping decentralized estimation with missing observations and communication faults. *Automatica*, 45(6), pp.1397 – 1406. (Cited on page 138.)

Stone, L.D., Streit, R.L., Corwin, T.L. and Bell, K.L., 2013. *Bayesian multiple target tracking*. Artech House. (Cited on page 26.)

Szafranski, G., Czyba, R., Janusz, W. and Blotnicki, W., 2013. Altitude estimation for the UAV's applications based on sensors fusion algorithm. *Proceedings of the International Conference on Unmanned Aircraft Systems*. pp.508–515. (Cited on page 150.)

Thomas, B., Holland, J.D. and Minot, E.O., 2012. Wildlife tracking technology options and cost considerations. *Wildlife Research*, 38(8), pp.653–663. (Cited on pages 2, 26, 62, 65, 92, and 96.)

Thrun, S., Burgard, W. and Fox, D., 2005. *Probabilistic robotics*. MIT press. (Cited on pages 95 and 100.)

Tian, X., Yuan, T. and Bar-Shalom, Y., 2015. Track-to-Track Fusion in Linear and Nonlinear Systems. *Advances in Estimation, Navigation, and Spacecraft Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.21–41. (Cited on page 122.)

Tokekar, P., Bhadauria, D., Studenski, A. and Isler, V., 2010. A robotic system for monitoring carp in Minnesota lakes. *Journal of Field Robotics*, 27(6), pp.779–789. (Cited on page 28.)

Tonissen, S.M. and Bar-Shalom, Y., 1998. Maximum likelihood track-before-detect with fluctuating target amplitude. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3), pp.796–809. (Cited on pages 64 and 74.)

Tremblay, J.A., Desrochers, A., Aubry, Y., Pace, P. and Bird, D.M., 2017. A low-cost technique for radio-tracking wildlife using a small standard unmanned aerial vehicle. *Journal of Unmanned Vehicle Systems*, 5(3), pp.102–108. (Cited on page 26.)

Udwani, R., 2018. Multi-objective maximization of monotone submodular functions with cardinality constraint. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. pp.9493–9504. (Cited on page 110.)

Üney, M., Clark, D.E. and Julier, S.J., 2013. Distributed fusion of PHD filters via exponential mixture densities. *IEEE Journal of Selected Topics in Signal Processing*, 7(3), pp.521–531. (Cited on pages 4, 121, and 138.)

Üney, M., Houssineau, J., Delande, E., Julier, S.J. and Clark, D.E., 2019. Fusion of finite-set distributions: pointwise consistency and global cardinality. *IEEE Transactions on Aerospace and Electronic Systems*, 55(6), pp.2759–2773. (Cited on page 121.)

Unmanned Systems Technology, 2019. Hybrid VTOL fixed-wing UAV manufacturers [Online]. Available from: `https://www.unmannedsystemstechnology.com/category/supplier-directory/platforms/hybrid-vtol-uav-manufacturers/` [Accessed 20 Feb 2020]. (Cited on page 150.)

Van Erven, T. and Harremos, P., 2014. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7), pp.3797–3820. (Cited on page 122.)

Vander Hook, J., Tokekar, P. and Isler, V., 2014. Cautious greedy strategy for bearing-only active localization: analysis and field experiments. *Journal of Field Robotics*, 31(2), pp.296–318. (Cited on page 28.)

Vo, B., Vo, B. and Beard, M., 2019. Multi-sensor multi-object tracking with the generalized labeled multi-bernoulli filter. *IEEE Transactions on Signal Processing*, 67(23), pp.5952–5967. (Cited on page 152.)

Vo, B.N., Mallick, M., Bar-Shalom, Y., Coraluppi, S., Osborne III, R., Mahler, R. and Vo, B.T., 2015. Multitarget tracking. *Wiley Encyclopedia of Electrical and Electronics Engineering*. (Cited on page 13.)

Vo, B.N., Singh, S. and Doucet, A., 2005. Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4), pp.1224–1245. (Cited on pages 15 and 63.)

Vo, B.N., Vo, B.T. and Hoang, H.G., 2016. An efficient implementation of the generalized labeled multi-Bernoullifilter. *IEEE Transactions on Signal Processing*, 65(8), pp.1975–1987. (Cited on pages 139 and 140.)

Vo, B.N., Vo, B.T., Pham, N.T. and Suter, D., 2010. Joint detection and estimation of multiple objects from image observations. *IEEE Transactions on Signal Processing*, 58(10), pp.5129–5141. (Cited on pages 17, 64, 75, and 76.)

Vo, B.N., Vo, B.T. and Phung, D., 2014. Labeled random finite sets and the Bayes multi-target tracking filter. *IEEE Transactions on Signal Processing*, 62(24), pp.6554–6567. (Cited on pages 17, 18, 81, 121, 122, 123, and 128.)

Vo, B.T., See, C.M., Ma, N. and Ng, W.T., 2012. Multi-sensor joint detection and tracking with the Bernoulli filter. *IEEE Transactions on Aerospace and Electronic Systems*, 48(2), pp.1385–1402. (Cited on pages 92 and 96.)

Vo, B.T. and Vo, B.N., 2013. Labeled random finite sets and multi-object conjugate priors. *IEEE Transactions on Signal Processing*, 61(13), pp.3460–3475. (Cited on pages 16, 18, 23, 79, and 121.)

Vo, B.T., Vo, B.N. and Cantoni, A., 2009. The cardinality balanced multi-target multi-Bernoulli filter and its implementations. *IEEE Transactions on Signal Processing*, 57(2), pp.409–423. (Cited on pages 18 and 121.)

VonEhr, K., Hilaski, S., Dunne, B.E. and Ward, J., 2016. Software defined radio for direction-finding in UAV wildlife tracking. *Proceedings of the IEEE International Conference on Electro Information Technology*. pp.0464–0469. (Cited on pages 28, 54, 55, 56, 57, and 150.)

Wai, H.T., Yang, Z., Wang, P.Z. and Hong, M., 2018. Multi-agent reinforcement learning via double averaging primal-dual optimization. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. pp.9649–9660. (Cited on pages 4 and 92.)

Wang, B., Yi, W., Hoseinnezhad, R., Li, S., Kong, L. and Yang, X., 2016. Distributed fusion with multi-Bernoulli filter based on generalized covariance intersection. *IEEE Transactions on Signal Processing*, 65(1), pp.242–255. (Cited on pages 4 and 121.)

Wang, X., Hoseinnezhad, R., Gostar, A.K., Rathnayake, T., Xu, B. and Bab-Hadiashar, A., 2018. Multi-sensor control for multi-object Bayes filters. *Signal Processing*, 142, pp.260–270. (Cited on pages 18, 63, 93, and 94.)

Ward, S., Hensler, J., Alsalam, B. and Gonzalez, L.F., 2016. Autonomous UAVs wildlife detection using thermal imaging, predictive navigation and computer vision. *Proceedings of the IEEE Aerospace Conference*. pp.1–8. (Cited on page 26.)

Webber, D., Hui, N., Kastner, R. and Schurgers, C., 2017. Radio receiver design for unmanned aerial wildlife tracking. *Proceedings of the International Conference on Computing, Networking and Communications*. pp.942–946. (Cited on page 26.)

Whiteson, S. and Roijers, D.M., 2016. Tutorial: multi-objective planning under uncertainty. *Proceedings of the 26th International Conference on Automated Planning and Scheduling*. (Cited on page 102.)

Wikelski, M., s, R.W., Kasdin, N.J., Thorup, K., Smith, J.A. and Swenson, G.W., 2007. Going wild: what a global small-animal tracking system could do for experimental biologists. *Journal of Experimental Biology*, 210(2), pp.181–186. (Cited on page 26.)

Yi, W., Li, S., Wang, B., Hoseinnezhad, R. and Kong, L., 2020. Computationally efficient distributed multi-sensor fusion with multi-Bernoulli filter. *IEEE Transactions on Signal Processing*, 68, pp.241–256. (Cited on pages 121 and 122.)

Zhou, D., 2013. *Thermal image-based deer detection to reduce accidents due to deer-vehicle collisions* [Online]. University of Minnesota, Intelligent Transportation Systems

Institute, Center for Transportation Studies. Available from: http://hdl.handle.net/11299/144870 [Accessed 1 Sep 2018]. (Cited on page 26.)

Zhu, Y., Wang, J. and Liang, S., 2019. Multi-objective optimization based multi-Bernoulli sensor selection for multi-target tracking. *Sensors*, 19(4), pp.980 – 997. (Cited on page 94.)

# Biography

**Hoa Van Nguyen** received his Bachelor degree in electrical engineering from Portland State University, Oregon, U.S.A in 2012. He worked as a process engineer and data science engineer at Intel Products Vietnam, Ho Chi Minh City, Vietnam from 2012-2016.

In 2016, he was awarded an Adelaide Scholarships International to pursue his doctoral degree at the School of Computer Science, the University of Adelaide under the supervision of Associate Professor Damith Chinthana Ranasinghe and Doctor Seyed Hamid Rezatofighi. His research interests include multi-object tracking, signal processing, Bayesian filtering and estimation, and multi-sensor control. He is a graduate student member of the IEEE.

Hoa Van Nguyen
hoavan.nguyen@adelaide.edu.au