10-4-2013

Multiple-Target Tracking in Complex Scenarios

Srinivas Phani Kumar Chavali
Washington University in St. Louis

WASHINGTON UNIVERSITY IN ST. LOUIS

School of Engineering and Applied Science

Preston M. Green Department of Electrical & Systems Engineering

Dissertation Examination Committee:
Dr. Arye Nehorai, Chair
Dr. R. Martin Arthur
Dr. Humberto Gonzalez
Dr. Nan Lin
Dr. Hiro Mukai
Dr. Naveen Singla

Multiple-Target Tracking in Complex Scenarios

by

Srinivas Phani Kumar Chavali

A dissertation presented to the Graduate School of Arts and Sciences
of Washington University in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2013
Saint Louis, Missouri

Contents

List of Figures . . . . . **v**

List of Algorithms . . . . . **vii**

List of Notations . . . . . **viii**

Acknowledgments . . . . . **ix**

Abstract . . . . . **xi**

1 Introduction **1**
1.1 Mathematical Formulation of MTT 3
1.1.1 Kalman Filtering 5
1.1.2 Particle Filtering 7
1.2 Original Contributions 9
1.3 Organization 13

2 Sparsity based MTT in Multipath Environments **14**
2.1 Introduction 14
2.2 System Model 17
2.2.1 Multipath Environment Model 17
2.2.2 Signal Model 21
2.2.3 State-Space Model 22
2.2.4 Measurement Model 24
2.3 Sparse Modeling 25
2.4 Sparsity-Based Multiple-Target Tracking 31
2.4.1 Standard Sparse Signal Reconstruction Techniques 32
2.4.2 Effect of Multipath Environment 34
2.4.3 Projection-Based (PB) Support Recovery Algorithm 40
2.5 Numerical Results 43
2.6 Summary 46

3 Multiple Rao Blackwellized Particle Filtering for MTT **50**
3.1 Introduction 50
3.2 System Model 52
3.2.1 State-Space Model 53
3.2.2 Measurement Model 55

3.3 Multiple Rao-Blackwellized Particle Filter . . . 59
3.4 Antenna Scheduling and Power Allocation . . . 63
3.4.1 Computation of the Posterior Cramér Rao Bound . . . 63
3.4.2 Approximate Greedy Algorithm for Adaptive Antenna Scheduling and Power Allocation . . . 66
3.5 Numerical Results . . . 67
3.6 Summary . . . 76

4 Hierarchical Particle Filtering using Multi-Modal Sensors . . . 79
4.1 Introduction . . . 79
4.2 Existing Techniques for Data Fusion . . . 81
4.3 Hierarchical Bayesian Models . . . 84
4.4 Particle Filtering with Existing Fusion Methods . . . 85
4.4.1 Linear Opinion Pool . . . 85
4.4.2 Independent Opinion Pool . . . 87
4.4.3 Independent Likelihood Pool . . . 87
4.5 Hierarchial Particle Filter . . . 88
4.5.1 Sampling from the Proposal Distribution . . . 90
4.5.2 Evaluating the Likelihood . . . 92
4.5.3 Discussion . . . 93
4.5.4 Examples . . . 93
4.6 Multiple-Target Tracking using HPF . . . 94
4.6.1 System Model . . . 95
4.6.2 State-Space Model . . . 95
4.6.3 Measurement Models . . . 97
4.6.4 Filtering Algorithm . . . 102
4.7 Numerical Results . . . 103
4.8 Summary . . . 107

5 Multi-Modal Sensor Management using Price Theory . . . 112
5.1 Introduction . . . 112
5.2 Problem Description . . . 114
5.3 Price theory and Auctions: Preliminaries . . . 117
5.3.1 Walrasian Equilibrium . . . 117
5.3.2 Auctions and Price discovery . . . 121
5.4 Sensor Management for MTT . . . 125
5.5 Numerical Results . . . 130
5.6 Summary . . . 136

6 Concurrent Particle Filtering for MTT in the Presence of Association Errors . . . 138
6.1 Introduction . . . 138
6.1.1 Review of Related Work . . . 139

6.2 System Model . . . 142
6.2.1 State-Space Model . . . 143
6.2.2 Measurement Model . . . 144
6.3 Data Association using Monte-Carlo Methods . . . 146
6.3.1 Interacting Multi-Model Sequential Sampling Particle Filtering 148
6.3.2 Interacting Multi-Model Independent Partition Particle Filtering 153
6.4 Game theory Preliminaries . . . 154
6.4.1 Normal Form Game . . . 156
6.4.2 Nash Equilibrium . . . 156
6.4.3 Correlated Equilibrium . . . 157
6.5 Concurrent Data Association via Game Theory . . . 159
6.5.1 Data Association Game . . . 159
6.5.2 Utility Functions . . . 160
6.5.3 Correlated Equilibrium using Regret Matching . . . 161
6.5.4 Equilibrium Characterization of Data Association Game . . . 164
6.6 Interacting Multi-Model Concurrent Particle Filtering . . . 168
6.7 Numerical Results . . . 169
6.8 Summary . . . 176

7 Conclusions . . . 182
7.1 Summary of Contributions . . . 182
7.2 Future Directions . . . 185

Appendix A Proof of Theorem 4 . . . 187

Appendix B Evaluation of Partial Derivatives . . . 189

References . . . 191

Vita . . . 203

List of Figures

1.1 Graphical model representing the state-space and the measurement models. 4

2.1 Block diagram showing the forward, reverse transmission channels and the targets. 18
2.2 Root mean-square-error for various tracking algorithms. 45
2.3 Actual Vs estimated trajectories using various tracking algorithms. . 47
2.4 Upper bound on error probability for an optimal reconstruction. . . . 48
2.5 Root mean-square-error for various values of PQ. 49

3.1 Three configurations used in the numerical examples. 73
3.2 Average CRMSE of the range and the velocity estimates plotted against the number of particles for the SPF and the MRBPF. 73
3.3 Performance comparison with and without adaptive scheduling and power allocation for the second configuration. 76
3.4 Performance comparison with and without adaptive scheduling and power allocation for the third configuration. 77
3.5 Performance comparison with and without multipath modeling. . . . 77

4.1 RMSE for the three partitions for the first example. 108
4.2 RMSE for the three partitions for the second example. 109
4.3 Performance comparison for varying number of targets. 110
4.4 Performance comparison for fixed number of targets. 111
4.5 Estimated Vs Actual target trajectories. 111

5.1 Performance comparison of the two approaches for the first scenario. . 132
5.2 Performance comparison of the two approaches for the second scenario. 135
5.3 Solutions to the SS, RA, and DF problems for the first scenario. . . . 136
5.4 Solutions to the SS, RA, and DF problems for the second scenario. . 137

6.1 Estimated vs the actual trajectories of the targets obtained using different algorithms for (a), (d), (g), (j) easy, (b), (e), (h), (k) medium, and (c), (f), (i), (l) hard problems. 178
6.2 CMRSE in the state estimation of various algorithms for an easy problem. 179
6.3 CMRSE in the state estimation of various algorithms for medium problem. 179
6.4 CMRSE in the state estimation of various algorithms for hard problem. 179

6.5 TCRMSE of various algorithms as a function of number of particles for (a) easy, (b) medium and (c) hard problems. 180
6.6 ICAR of various algorithms as a function of number of particles for (a) easy, (b) medium and (c) hard problems. 180
6.7 ICMR of various algorithms as a function of number of particles for (a) easy, (b) medium and (c) hard problems. 180
6.8 ROC graph for various values of μ_2. 181
6.9 Probability distributions over the strategies as a function of the iteration number for (a) Target 1 (b) Target 2 and (c) Target 3. 181

List of Algorithms

1 PB Support Recovery Algorithm for Target Tracking in Multipath Environment 42
2 MRBPF Algorithm for Joint Tracking of Target and Channel States. 62
3 Greedy Algorithm for Sensor Scheduling and Power Allocation. 68
4 Auction Algorithm to find Walrasian Equilibrium of a Double-Sided Market 124
5 IMM-SSPF Algorithm for MTT with Association Ambiguities. 152
6 IMM-IPPF Algorithm for MTT with Association Ambiguities. 155
7 IMM-CPF Algorithm for MTT with Association Ambiguities. 169

List of Notations

a	lowercase math italic denotes a scalar
$\boldsymbol{a}$	lowercase bold denotes a vector $\boldsymbol{a} \in \mathbb{C}^n$
$\boldsymbol{A}$	uppercase bold denotes a matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$
$\lvert\boldsymbol{A}\rvert$	determinant of $\boldsymbol{A}$
$[\boldsymbol{A}]_{(i,j)}$	(i,j)-th element of $\boldsymbol{A}$
$\boldsymbol{A}^*$	conjugate of $\boldsymbol{A}$
$\boldsymbol{A}^T$	transpose of $\boldsymbol{A}$
$\boldsymbol{A}^H$	conjugate-transpose (Hermitian) of $\boldsymbol{A}$
$\boldsymbol{A}^-$	generalized inverse of $\boldsymbol{A}$ such that $\boldsymbol{A}\boldsymbol{A}^-\boldsymbol{A} = \boldsymbol{A}$
$\boldsymbol{A}^\dagger$	pseudo-inverse of $\boldsymbol{A}$, defined as $\boldsymbol{A}^\dagger = \left(\boldsymbol{A}^H\boldsymbol{A}\right)^{-1}\boldsymbol{A}^H$
$\mathrm{vec}(\boldsymbol{A})$	forms a column vector of length mn by stacking the columns of $\boldsymbol{A}$
$\mathrm{tr}(\boldsymbol{A})$	trace of square matrix $\boldsymbol{A} \in \mathbb{C}^{n \times n}$
$\mathrm{diag}(\cdots)$	forms a square matrix with non-zero entries only on the main diagonal
$\boldsymbol{I}_n$	identity matrix of dimension n
$\mathrm{Re}\{\cdot\}$	real part of a complex quantity
$\mathrm{Im}\{\cdot\}$	imaginary part of a complex quantity
$\lceil\cdot\rceil$	nearest integer greater than a fractional quantity
$\lVert\cdot\rVert_0$	number of non-zero elements in a vector
$\lVert\cdot\rVert_p$	p-th norm of a vector, $p = 1, 2$
$\lVert\cdot\rVert_\infty$	maximum of the absolute-valued elements in a vector
$\langle\,,\,\rangle$	inner-product operator
$\otimes$	Kronecker product operator
$\odot$	element-wise Hadamard product operator
$\mathcal{N}$	Gaussian distribution
$\mathbb{C}\mathcal{N}$	complex Gaussian distribution

Acknowledgments

I am sincerely grateful to my advisor, Dr. Arye Nehorai, for his guidance and support throughout my doctoral research. I thank him for leading me to interesting research topics and giving me the freedom to work on different problems. Apart from research, Dr. Nehorai has been a very kind personal mentor. I feel fortunate to be a part of his excellent research group.

I would like to thank my dissertation committee members, Dr. R. Martin Arthur, Dr. Nan Lin, Dr. Hiro Mukai, Dr. Humberto Gonzalez and Dr. Naveen Singla, for their time and for providing constructive suggestions to improve this dissertation.

I am truly thankful to all my teachers and Professors so far, including Dr. V.U. Reddy, Dr. P.R.K. Rao, and Dr. J. Sivaswamy from IIIT-H, and Dr. J. A. O'Sullivan, Dr. H. Schattler, Dr. K. Weinberger, Dr. N. Lin, and Dr. Jr-Shin Li from WashU, for helping me to build a strong background in statistical signal processing.

I would also like convey my heartiest thanks to all my labmates for their friendship and for making work more enjoyable in lab. The thoughtful discussions we shared helped me understand several different perspectives of a problem.

I would like to thank my family and all my friends (the list is long), both back home and here in St. Louis for their support over the last few years. My special thanks go to my wife Soumya for her support, encouragement, and persistent confidence in me, especially during the final stages of this dissertation writing. I wouldn't have been half as calm if it weren't for her. Her unconditional love made the last two years most special for me. Lastly, my parents and my brother deserve a special mention for their support. I offer my deepest gratitude to my parents, Prasad and Ratna, for their endless care, innumerable prayers and sacrifices, and their encouragement which provided me with an immense inspiration.

Srinivas Phani Kumar Chavali

Washington University in Saint Louis
December 2013

to my parents

ABSTRACT OF THE DISSERTATION

Multiple-Target Tracking in Complex Scenarios

by

Srinivas Phani Kumar Chavali

Doctor of Philosophy in Electrical Engineering

Washington University in St. Louis, December 2013

Professor Arye Nehorai, Chair

In this dissertation, we develop computationally efficient algorithms for multiple-target tracking (MTT) in complex scenarios. For each of these scenarios, we develop measurement and state-space models, and then exploit the structure in these models to propose efficient tracking algorithms. In addition, we address design issues such as sensor selection and resource allocation.

First, we consider MTT when the targets themselves are moving in a time-varying multipath environment. We develop a sparse-measurement model that allows us to exploit the inherent joint delay-Doppler diversity offered by the environment. We then reformulate the problem of MTT as a block-support recovery problem using the sparse measurement model. We exploit the structure of the dictionary matrix to develop a computationally efficient block support recovery algorithm (and thereby a multiple-target tracking algorithm) under the assumption that the channel state describing the time-varying multipath environment is known. Further, we also derive an upper bound on the overall error probability of wrongly identifying the support of the sparse

signal. We then relax the assumption that the channel state is known. We develop a new particle filter called the Multiple Rao-Blackwellized Particle Filter (MRBPF) to jointly estimate both the target and the channel states. We also compute the posterior Cramér-Rao bound (PCRB) on the estimates of the target and the channel states and use the PCRB to find a suitable subset of antennas to be used for transmission in each tracking interval, as well as the power transmitted by these antennas.

Second, we consider the problem of tracking an unknown number and types of targets using a multi-modal sensor network. In a multi-modal sensor network, different quantities associated with the same state are measured using sensors of different kinds. Hence, an efficient method that can suitably combine the diverse information measured by each sensor is required. We first develop a Hierarchical Particle Filter (HPF) to estimate the unknown state from the multi-modal measurements for a special class of problems which can be modeled hierarchically. We then model our problem of tracking using a hierarchical model and then use the proposed HPF for joint initiation, termination and tracking of multiple targets. The multi-modal data consists of the measurements collected from a radar, an infrared camera and a human scout. We also propose a unified framework for multi-modal sensor management that comprises sensor selection (SS), resource allocation (RA) and data fusion (DF). Our approach is inspired by the trading behavior of economic agents in commercial markets. We model the sensors and the sensor manager as economic agents, and the interaction among them as a double sided market with both consumers and producers. We propose an iterative double auction mechanism for computing the equilibrium of such a market. We relate the equilibrium point to the solutions of SS, RA and DF.

Third, we address MTT problem in the presence of data association ambiguity that arises due to clutter. Data association corresponds to the problem of assigning a

measurement to each target. We treat the data association and state estimation as separate subproblems. We develop a game-theoretic framework to solve the data association, in which we model each tracker as a player and the set of measurements as strategies. We develop utility functions for each player, and then use a regret-based learning algorithm to find the correlated equilibrium of this game. The game-theoretic approach allows us to associate measurements to all the targets simultaneously. We then use particle filtering on the reduced dimensional state of each target, independently.

Chapter 1

Introduction

Multiple target tracking (MTT) [1], [2], [3], [4], [5] first originated in the 1970s, and since then it has been one of the most challenging research areas with applications in many critical real-world tasks. In the initial years, the research was driven by military applications such as estimating the positions and speeds of the airborne vehicles using ground radars. Later, as the research in the area evolved, tools developed for the original application were extended to several other domains including image processing [6], [7], video surveillance [8], mobile robotics [9], [10], finance [11], [12] and biomedicine [13].

Sensor systems, such as radar, lidar or a camera, collect noisy *measurements* originating from the objects of interest, referred to as *targets* (sometimes the measurements can originate from clutter or spurious detections). In a military task, the targets correspond to the airborne vehicles or missiles and the measurement collected by a radar corresponds to a back-propagated and filtered electromagnetic waveform. The problem of MTT is to then use a series of these measurements and estimate the quantities of interest usually referred to as the *state* of the target. In the traditional military

setting, the state corresponds to the positions and velocities of the moving vehicles. MTT can thus be thought of as a recursive estimation problem over time.

MTT comprises two steps: (a) prediction and (b) update. In order to predict the current state using an estimate (of the state) from the past, a process model that describes the state evolution is needed. This process model is known as the state-space model. Typically, the state-space model is described using a first-order Markov process, and it relates the current state to the previous state. A measurement model describes the relationship between the current state and the current observation. Once the measurements are available, the predicted value of the state is updated using these measurements. Typically, a single sensor is used to obtain the measurements of the targets. Sometimes, sensors of a single type may not reliably obtain information about the entire state. In such cases, several different kinds of sensors are employed to obtain information about the state. Such measurements are referred to as multi-modal measurements.

The problem of MTT can become challenging due to several reasons such as targets not being detected by the sensor or the targets moving in a dense urban environments. In this dissertation, we develop computationally efficient tracking algorithms for multiple-target tracking in three complex scenarios: (i) targets moving in a time-varying multipath environment; (ii) unknown and time-varying number and types of targets; (iii) targets with low probability of detection moving in dense clutter. We focus on a class of algorithms called the sequential Monte-Carlo filters or the particle filters for addressing MTT. The MTT problem in this dissertation concerns tracking independent point targets. A point target is modelled in a dynamical state-space usually consisting of position, velocity in the two-dimensional space. The tracking

problem then corresponds to estimation of the target state given the noisy measurements from a sensor whose field of view includes the targets of interest. In the next section, we formulate MTT as a sequential Bayesian estimation problem and briefly discuss the ideas of Kalman and particle filters. We will then summarize the contributions of this dissertation.

1.1 Mathematical Formulation of MTT

In this section, we formulate MTT as a sequential (or recursive) Bayesian estimation problem also known as *Bayesian filtering*. A detailed survey in the area of Bayesian filtering is provided in [14].

Let $\boldsymbol{\theta}_t$ denote the state of a target at time t. For notational simplicity, assume that there is only one target in the region of interest. We extend the filtering algorithms that will be discussed here to the case of multiple-targets in the subsequent chapters. Typically, the state $\boldsymbol{\theta}_t$ of the target comprises its x-position, y-position, x-velocity and y-velocity, i.e., $\boldsymbol{\theta}_t = [x_t, y_t, \dot{x}_t, \dot{y}_t]^T$. However, the state can also be comprised of other target or system parameters as we will see in the subsequent chapters.

A state-space model describes the evolution of the target over time. Formally, the state-space model is described as:

$$\boldsymbol{\theta}_t = g(\boldsymbol{\theta}_{t-1}, \boldsymbol{v}_{t-1}). \tag{1.1}$$

In the above, $g(\cdot)$ is a known, but possibly nonlinear function and $\boldsymbol{v}_t$ is the stochastic process noise. In addition, there are noisy measurements $\boldsymbol{y}_t$, which depend on the

state $\boldsymbol{\theta}_t$ in some known manner via a measurement model:

$$\boldsymbol{y}_t \;=\; h(\boldsymbol{\theta}_t, \boldsymbol{w}_t), \tag{1.2}$$

In the above, $h(\cdot)$ is a known, but possibly nonlinear measurement function and $\boldsymbol{w}_t$ is the stochastic measurement noise at time t. The state equation and the measurement equation can be equivalently written in terms of the conditional probability distributions, $p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1})$ and $p(\boldsymbol{y}_t \mid \boldsymbol{\theta}_t)$. Note that the measurement $\boldsymbol{y}_t$ at time t is independent of the past states and measurements, given the current state $\boldsymbol{\theta}_t$. The state equation and the measurement equation can be represented as a graph as shown in Fig. 1.1.

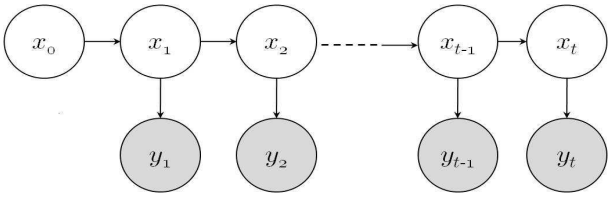


Figure 1.1: Graphical model representing the state-space and the measurement models.

Given a system characterized by a state-space model and a measurement model, our goal is to estimate the posterior distribution of the unknown state vector $\boldsymbol{\theta}_t$, $t \in \mathbb{N}$, given the past measurements $\boldsymbol{y}_{1:t-1} = \{\boldsymbol{y}_1, \boldsymbol{y}_2 \cdots \boldsymbol{y}_{t-1}\}$ and the current measurement $\boldsymbol{y}_t$. Under the sequential Bayesian framework, the posterior of the state given the measurements can be computed using the Chapman-Kolmogorov equation and Bayes' theorem [15]:

$$p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}) p(\boldsymbol{\theta}_{t-1} \mid \boldsymbol{y}_{1:t-1}) \mathrm{d}\boldsymbol{\theta}_{\mathrm{t-1}}, \text{ and} \tag{1.3}$$

$$p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t}) = \frac{1}{z}p(\boldsymbol{y}_t \mid \boldsymbol{\theta}_t)p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{1:t-1}), \tag{1.4}$$

where z is a normalization constant. Equations (1.3), and (1.4) are called the time-update equation and the measurement-update equation, respectively.

1.1.1 Kalman Filtering

If the state model and the measurement model are linear, and the noise processes are additive Gaussian, the Kalman Filter (KF) [15] is an efficient and optimal solution to the Bayesian filtering problem in the least-squares sense. Suppose that the state-space model and the measurement model given in Eqs (1.1) and (1.2), respectively are simplified to:

$$\boldsymbol{\theta}_t = \boldsymbol{F}_{t-1}\boldsymbol{\theta}_{t-1} + \boldsymbol{v}_{t-1}, \tag{1.5}$$

$$\boldsymbol{y}_t = \boldsymbol{H}_t\boldsymbol{\theta}_t + \boldsymbol{w}_t, \tag{1.6}$$

where $\boldsymbol{v}_t$ and $\boldsymbol{w}_t$ are independently distributed as

$$\boldsymbol{v}_t \sim \mathcal{N}(\boldsymbol{v}_t; \mathbf{0}, \boldsymbol{\Sigma}_{\mathrm{v}}), \tag{1.7}$$

$$\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{w}_t; \mathbf{0}, \boldsymbol{\Sigma}_{\mathrm{w}}). \tag{1.8}$$

A Kalman filter for obtaining an estimate of the posterior distribution of $\boldsymbol{\theta}_t$ is then summarized by the following equations:

$$\begin{aligned}
\boldsymbol{\theta}_{t|t-1} &= \boldsymbol{F}_{t-1}\boldsymbol{\theta}_{t-1|t-1} && (1.9)\\
\boldsymbol{P}_{t|t-1} &= F_{t-1}\boldsymbol{P}_{t-1|t-1}\boldsymbol{F}_{t-1}^{T} + \boldsymbol{\Sigma}_{\mathrm{v}} && (1.10)\\
\tilde{\boldsymbol{y}}_t &= \boldsymbol{y}_t - \boldsymbol{H}_t\boldsymbol{\theta}_{t|t-1} && (1.11)\\
\boldsymbol{S}_t &= \boldsymbol{H}_t\boldsymbol{P}_{t|t-1}\boldsymbol{H}^{T} + \boldsymbol{\Sigma}_{\mathrm{w}} && (1.12)\\
\boldsymbol{K}_t &= \boldsymbol{P}_{t|t-1}\boldsymbol{H}_t^{T}\boldsymbol{S}_t^{-1} && (1.13)\\
\boldsymbol{\theta}_{t|t} &= \boldsymbol{\theta}_{t|t-1} + \boldsymbol{K}_t\tilde{\boldsymbol{y}}_t && (1.14)\\
\boldsymbol{P}_{t|t} &= (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{H}_t)\boldsymbol{P}_{t|t-1} && (1.15)
\end{aligned}$$

In the above, Eqs (1.9) and (1.10) correspond to the time-update and Eqs. (1.11)-(1.15) correspond to the measurement update. $\boldsymbol{K}_t$ is known as the Kalman gain. The posterior distribution $p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{1:t})$ is then given as:

$$p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{1:t}) \sim \mathcal{N}(\boldsymbol{\theta}_t; \boldsymbol{\theta}_{t|t}, \boldsymbol{P}_{t|t}). \quad (1.16)$$

If the state-space and the measurement models are nonlinear, Unscented Kalman filter (UKF) [16], Extended Kalman Filter (EKF) [17] can be used for Bayesian filtering. However, in many practical problems linearizations and Gaussian approximations are intractable or yield low performance. A sequential Monte-Carlo (SMC) filter, also called the particle filter (PF) provides a general framework for Bayesian filtering for non-linear and non-Gaussian models.

1.1.2 Particle Filtering

Particle filtering [18], [19], [20] is a Monte Carlo approach to address the Bayesian filtering, and particle filters (PFs) provide a tractable solution to the state estimation problem in non-linear and non-Gaussian systems. A PF approximates the joint posterior distribution of the state (up to the current time) using a set of particles, and associated weights,

$$p(\boldsymbol{\theta}_{0:t} \mid \boldsymbol{y}_{1:t}) \approx \sum_{i=1}^{N_\mathrm{s}} w_t^{(i)} \delta(\boldsymbol{\theta}_{0:t} - \boldsymbol{\theta}_{0:t}^{(i)}), \tag{1.17}$$

where $\left\{\boldsymbol{\theta}_{0:t}^{(i)}\right\}_{i=1}^{N_\mathrm{s}}$ are the particles (or samples) that characterize the probability distribution $p(\boldsymbol{\theta}_{0:t}|\boldsymbol{y}_{1:t})$, and $\left\{w_t^{(i)}\right\}_{i=1}^{N_\mathrm{s}}$ are the associated weights. The samples $\left\{\boldsymbol{\theta}_{0:t}^{(i)}\right\}_{i=1}^{N_\mathrm{s}}$ are drawn from a known proposal distribution, and the weights are derived using the principle of importance sampling [21]. According to this principle, the weights are updated using

$$\tilde{w}_t^{(i)} = \frac{p(\boldsymbol{\theta}_{0:t}^{(i)} \mid \boldsymbol{y}_{1:t})}{q(\boldsymbol{\theta}_{0:t}^{(i)} \mid \boldsymbol{y}_{1:t})}, \tag{1.18}$$

where $q(\boldsymbol{\theta}_{0:t} \mid \boldsymbol{y}_{1:t})$ is the proposal distribution from which the samples are drawn and $\tilde{w}_t^{(i)}$ is the un-normalized weight of the i^{th} particle at time t. The weight update equation can be simplified as:

$$\begin{aligned}
\tilde{w}_t^{(i)} &\propto \frac{p(\boldsymbol{\theta}_{0:t}^{(i)} \mid \boldsymbol{y}_{1:t})}{q(\boldsymbol{\theta}_{0:t}^{(i)} \mid \boldsymbol{y}_{1:t})} \\
&\propto \frac{p(\boldsymbol{\theta}_{0:t-1}^{(i)} \mid \boldsymbol{y}_{1:t-1})}{q(\boldsymbol{\theta}_{0:t-1}^{(i)} \mid \boldsymbol{y}_{1:t-1})} \times p(\boldsymbol{y}_t \mid \boldsymbol{\theta}_t^{(i)}) \times \frac{p(\boldsymbol{\theta}_t^{(i)} \mid \boldsymbol{\theta}_{t-1}^{(i)})}{q(\boldsymbol{\theta}_t^{(i)} \mid \boldsymbol{\theta}_{0:t-1}^{(i)}, \boldsymbol{y}_{1:t})} \\
&\propto w_{t-1}^{(i)} \times p(\boldsymbol{y}_t \mid \boldsymbol{\theta}_t^{(i)}) \times \frac{p(\boldsymbol{\theta}_t^{(i)} \mid \boldsymbol{\theta}_{t-1}^{(i)})}{q(\boldsymbol{\theta}_t^{(i)} \mid \boldsymbol{\theta}_{0:t-1}^{(i)}, \boldsymbol{y}_{1:t})} \\
&\propto w_{t-1}^{(i)} \times p(\boldsymbol{y}_t \mid \boldsymbol{\theta}_t^{(i)}),
\end{aligned} \tag{1.19}$$

where the proposal distribution $q(\boldsymbol{\theta}_{0:t} \mid \boldsymbol{y}_{1:t})$ is chosen to factorize such that

$$q(\boldsymbol{\theta}_{0:t} \mid \boldsymbol{y}_{1:t}) = q(\boldsymbol{\theta}_{0:t-1} \mid \boldsymbol{y}_{1:t-1}) \times q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{0:t-1}, \boldsymbol{y}_{1:t}), \tag{1.20}$$

and $q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{0:t-1}, \boldsymbol{y}_{1:t}) = q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}, \boldsymbol{y}_t)$. Due to this choice, the proposal distribution only depends on $\boldsymbol{\theta}_{t-1}$, and we can discard the samples $\boldsymbol{\theta}_{0:t-1}$, and the observations $\boldsymbol{y}_{1:t-1}$. The last step of Eq. (1.19) is obtained by choosing the proposal distribution to be the state transition distribution

$$q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}, \boldsymbol{y}_t) = p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}). \tag{1.21}$$

Although this choice is not optimal, it results in a simple weight update equation. Hence the state transition distribution is a most popular choice for the proposal distribution.

Resampling

Standard particle filters, based on the principle of importance sampling, suffer from a drawback called the degeneracy phenomenon. After a few iterations, the weights of all but a few particles will be close to zero. As a result of degeneracy, the number of particles contributing to the posterior distribution become significantly less over time, and hence the performance of the filter degrades. In theory, it is impossible to avoid degeneracy, but its effect can be reduced by drawing samples from a *good* proposal distribution and by using resampling techniques.

The idea of resampling is to eliminate the particles that have smaller weights and replace them with new samples with equal weights. Resampling is a crucial step

in particle filtering, since after a few iterations, propagating trivial weights becomes computationally expensive. After each iteration, the effective number of particles is computed using [18]:

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N_s} (w_t^{(i)})^2}, \tag{1.22}$$

and the particle set is resampled if N_{eff} falls below a predetermined threshold. There are several resampling techniques available in the literature; interested readers can refer [14]. In the rest of this dissertation, we use the residual systematic resampling technique that is described in [22].

In addition to the degeneracy, the number of particles required to approximate the posterior density grows exponentially [23] with the dimension of the state vector. Filters using such a large number of particles are computationally complex and run into numerical inconsistencies. One of the most common ways to overcome this difficulty is to partition the state-space into several subspaces, and explore each subspace independently [24].

1.2 Original Contributions

In this dissertation, we develop computationally efficient algorithms for MTT in complex scenarios. Below, we briefly summarize our contributions.

Sparsity-based Algorithm for MTT in Multipath Environments: We propose a sparsity-based algorithm for tracking targets moving in a time-varying multipath environment. The channel state that describes the multipath environment is assumed

to be known. We develop a sparse measurement model for the received signal, by considering a finite dimensional representation of the system function that characterizes the multipath environment. This measurement model allows us to exploit the inherent joint delay-Doppler diversity offered by the environment. We then reformulate the problem of MTT as a block-support recovery problem from the sparse measurement model. We prove that under some signaling conditions, the dictionary of the sparse measurement model exhibits a special structure. We exploit this structure to develop a computationally inexpensive block support recovery algorithm (and thereby a multiple-target tracking algorithm) by projecting the received signal onto the row space of the dictionary. Further, we also derive an upper bound on the overall error probability of wrongly identifying the support of the sparse signal.

Multiple Rao Blackwellized Particle Filter for MTT: We develop a particle filter for jointly estimating both the target and the channel states (when the channel state is unknown to the radar). The dimension of the overall state is high, and hence the standard particle filter requires a large number of particles making it computationally expensive. However, given the target state the measurements are linear and Gaussian in the channel state and the target states evolve independently of each other. We exploit this structure in the system to develop a new filter called the Multiple Rao-Blackwellized Particle Filter (MRBPF) to jointly estimate both the target and the channel state. MRBPF is a hybrid filter that uses one particle filter for each target's state partition, and uses a Kalman filter to analytically find an estimate of the channel state. Since MRBPF operates on low dimensional subspaces, it reduces the complexity involved with high-dimensional state space. In addition, due to the Rao-Blackwell theorem, the variance of the estimates obtained after finding the

channel state analytically will be less than the variance of the estimate obtained by sampling the entire state vector.We also compute the posterior Cramér-Rao bound (PCRB) on the estimates of the target state and the channel state. Using PCRB as a metric, we find a suitable subset of antennas to be used for transmission in each tracking interval, as well as the power transmitted by these antennas.

Hierarchical Particle Filter for Tracking Unknown Number of Targets using Multi-Modal Sensors: In a multi-modal sensor network, different quantities associated with the same state are measured using sensors of different kinds. We first develop a general filtering technique to estimate the unknown state from the multi-modal measurements for a special class of problems which can be modeled hierarchically. The proposed filter, the Hierarchical Particle Filter (HPF), estimates the global posterior density of the unknown state in multiple stages, by partitioning the state and the measurement spaces into lower dimensional subspaces. At each stage, we find an estimate of one partition using the measurements from the corresponding partition, and the information from the previous stages. In this way, the proposed filter can combine the information from different sensors. We then model the problem of MTT using a hierarchical model and demonstrate the proposed filtering for joint initiation, termination and tracking of an unknown number of targets. The multi-modal data consists of the measurements collected from a radar, an infrared camera and a human scout.

Sensor Management for MTT: We focus on sensor selection (SS), resource allocation (RA) and data fusion (DF), which constitute the sensor management. Our

approach is inspired by the trading behavior of economic agents in commercial markets. Each sensor node (SN) acts as a seller who wants to sell the data it collects, to the sensor network manager (SM) who acts as a buyer. The resources and the data are priced by looking to balance global supply and demand, with the SN required to purchase resources for producing the data, and the SM required to purchase data to accomplish his tasks. We model this interaction as a double sided market with both consumers and producers, and propose an iterative double auction mechanism for computing the equilibrium of such a market. We relate the equilibrium point to the solutions of SS, RA and DF.

Concurrent Particle Filter for MTT in the Presence of Association Ambiguities: Data association corresponds to the problem of assigning each measurement to a target or clutter. We propose a new particle filter, which we refer to as the interacting multi-model concurrent particle filter (IMM-CPF) to track multiple maneuvering targets in the presence of association ambiguities. We treat the data association and state estimation as separate subproblems and we develop a game-theoretic framework to address the data association, in which we model each tracker as a player and the set of measurements as strategies. We develop utility functions for each player, and then use a regret-based learning algorithm to find the equilibrium of this game. The game-theoretic approach allows us to associate measurements to all the targets simultaneously. We then use particle filtering on the reduced dimensional state of each target, independently.

1.3 Organization

The rest of dissertation is organized as follows. In **Chapter 2**, we address MTT when the targets themselves are moving in a time-varying multipath environment and propose a sparsity-based MTT algorithm. In **Chapter 3**, we relax the assumption that channel state describing time-varying multipath environment is known and propose MRBPF that jointly tracks both the channel and the target states. In **Chapter 4**, we consider the problem of tracking an unknown number and types of targets using a multi-modal sensor network, and we propose HPF. In **Chapter 5**, we develop a unified framework for sensor management for tracking using a multi-modal sensors. In **Chapter 6**, we address MTT in the presence of data association ambiguity that arises due to clutter and propose IMM-CPF. In **Chapter 7**, we summarize the conclusions reached in this thesis and provide several potential directions for further work in this area.

Chapter 2

Sparsity based MTT in Multipath Environments[1]

In this chapter, we address the problem of MTT in a time-varying multipath environment. Multipath environments are usually encountered when the targets are moving in dense urban environments.

2.1 Introduction

Multipath environments provide two major challenges for the operation of conventional radar systems [25]: complete obscuration of the target due to tall buildings; and the presence of multiple scatterers in the environment. Radar systems operating in these environments therefore suffer severe performance degradation compared to the conventional radar systems designed for the line-of-sight (LOS) propagation environments. In order to overcome the problem of obscuration, radars generally

[1]Based on P. Chavali and A. Nehorai, "A Low-Complexity Multi-Target Tracking Algorithm for Urban Environments using Sparse Modeling", *Signal Processing*, vol. 92, pp. 2199-2213, Sep. 2012.

maintain a steep grazing angle. This, however, dramatically decreases the coverage area [25]. The degradation due to presence of scatterers can be compensated by treating the signals arriving on the non-LOS paths as interference, and thus mitigating it [26], [27]. Recently, however, there has been a growing interest in exploiting the multipath nature of the environment to obtain a better performance [28], [29]. The presence of multiple scatterers in the environment introduces a delay spread in the transmission channel. The relative motion between the moving targets and scatterers introduces time variations in the channel, which manifests as Doppler spread. Thus multipath environments with multiple scatterers are characterized by a time-varying multipath channel. The delay and the Doppler spread provide additional diversity for the problem of target tracking, and by employing methods that exploit this diversity, the performance of the tracking system can be significantly improved.

In this chapter, a finite-dimensional measurement model is developed for the tracking problem by modeling multipath channel as a time-varying linear system. This measurement model captures both the delay and the Doppler spread that is introduced into the system. The dimensionality of the linear representation has a physical significance and it represents the additional diversity that the environment offers.

Next, sparse modeling is used to transform the problem of multiple-target tracking into a problem of support recovery of a block- sparse signal. Sparse modeling has drawn a lot of attention in the radar community [30] recently. In the context of radar, sparse models are used to reconstruct the target scene by representing each target as a grid point in the delay-Doppler plane. Since the number of targets is in general less than the number of grid points, the target scene can be represented as

a sparse signal and it can be reconstructed by finding the support [2] [31]- [32] of the sparse signal. Further, an upper bound on the overall error probability of wrongly identifying the support of the sparse target scene is derived. Using this bound, it will be shown that spread-spectrum waveforms are ideal candidates for signaling, as they have good time-bandwidth properties which are essential to obtain the full diversity provided by the delay and the Doppler spread.

Further, a computationally inexpensive algorithm for the support recovery of the sparse signal is introduced in this chapter. This method works by projecting the received signal vector onto the row space of the dictionary [3]. The dictionary corresponding to the block-sparse model consists of delayed and Doppler shifted versions of the transmitted waveforms. Under spread-spectrum signaling, we prove that the dictionary of the sparse model exhibits a special structure, which enables efficient support recovery. We refer to this algorithm as projection-based (PB) support recovery. We use PB support recovery algorithm for target tracking and we demonstrate using numerical simulations that PB algorithm takes significantly less time compared to the time taken by the standard sparse signal reconstruction based tracking methods, while giving good performance.

[2]Support of a vector $\boldsymbol{x}$ is defined as the set of indices where the vector is non-zero, i.e., $\text{supp}(\boldsymbol{x}) = \{i \,\big|\, x_i \neq 0\}$

[3]The dictionary of a sparse model corresponds to the over complete set of basis functions obtained by discretizing the variable of interest.

2.2 System Model

2.2.1 Multipath Environment Model

Consider a monostatic pulse radar system operating in an urban environment as shown in Fig. 2.1. Assume that the radar antenna is omnidirectional. This assumption ensures that the antenna transmits equal power in all the directions and is capable of receiving the signal from all the directions. The radar transmits a known electromagnetic signal to obtain an image of the target. The transmitted signal propagates through a forward transmission channel before it reaches the target. The target responds to the transmitted waveform and backscatters a modified waveform into the environment. The modified waveform propagates back to the radar receiver through a reverse transmission channel. The received signal is then processed to obtain the necessary information about the target. This is a standard model for radar systems operating in a multipath environment [29], [33].

In the absence of the time-varying multipath effect due to the surrounding environment, the forward transmission channel and the reverse transmission channel do not have any effect on the backscattered waveform, except possibly for a propagation loss. In such cases, the target is the object which causes the uncertainty in the received waveform. The electromagnetic signal that is reflected off the target is modified as a function of the target's relative position, velocity and it's physical characteristics, referred to as the radar cross section (RCS) [34]. If the target is assumed to be a point target, the signal reflected off it can be expressed as

$$r(t') \;=\; \alpha s(t'-\tau_{\mathrm{d}})e^{j2\pi\nu_{\mathrm{d}}t'}, \tag{2.1}$$

where $s(t')$ is the transmitted waveform, $r(t')$ is the backscattered waveform from the target, α is the unknown RCS, τ_{d} is the round trip delay, and ν_{d} is the Doppler shift in the frequency due to the relative motion between radar and the target. For simplicity, assume initially that there is only one target present in the region of interest. The extension to the case of multiple targets will be made in Sec. 2.2.4. Without the loss of generality, consider that the radar is located at $(0, 0)$. With this assumption,

$$\tau_{\mathrm{d}} = \frac{2}{c}\sqrt{x^2 + y^2}, \tag{2.2}$$

$$\nu_{\mathrm{d}} = \frac{2f_{\mathrm{c}}}{c}\frac{x\dot{x} + y\dot{y}}{\sqrt{x^2 + y^2}}, \tag{2.3}$$

where (x, y) and $(\dot{x}, \dot{y})$ denote the position and velocity of the target, respectively, in the two dimensional plane, f_{c} is the carrier frequency, and c is the speed of propagation.

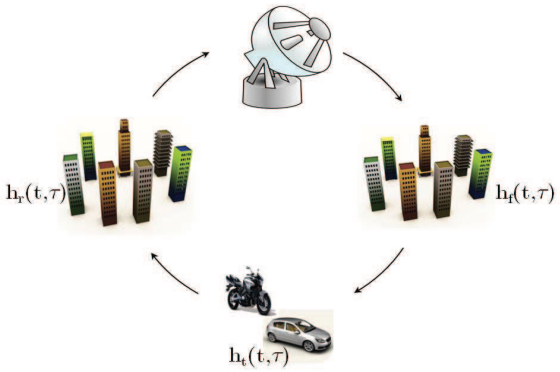


Figure 2.1: Block diagram showing the forward, reverse transmission channels and the targets.

In the presence of time-varying multipath channel, the uncertainty in the received waveform is due to the joint effect of the forward transmission channel, the target itself, and the reverse transmission channel. Since the propagation of electromagnetic waves obey the superposition principle, it is reasonable to model the forward

transmission channel and the reverse transmission channel as linear time-varying systems [33]. Denoting the time-varying system responses of the forward transmission channel and the reverse transmission channel using $h_{\mathrm{f}}(t', \tau)$ and $h_{\mathrm{r}}(t', \tau)$, respectively, the received signal can be expressed as

$$\begin{aligned} y(t') &= h_{\mathrm{r}}(t', \tau) * r(t') * h_{\mathrm{f}}(t', \tau) + w(t'), \\ &= \int h(t', \tau) r(t' - \tau) \mathrm{d}\tau + w(t'), \end{aligned} \tag{2.4}$$

where $y(t')$ is the received waveform in the presence of time-varying multipath channel, $h(t', \tau)$ is the response of the overall multipath channel at delay τ and time t' given as $h(t', \tau) = h_{\mathrm{r}}(t', \tau) * h_{\mathrm{f}}(t', \tau)$, and $w(t')$ is circularly symmetric, complex additive white Gaussian noise. Considering the Fourier transform of $h(t', \tau)$, the signal $y(t')$ can be expressed as

$$y(t') = \int \int H(f, \tau) r(t' - \tau) e^{j2\pi f t'} \mathrm{d}\tau \mathrm{d}f + w(t'), \tag{2.5}$$

where $H(f, \tau)$ is the Fourier transform $h(t', \tau)$ at time τ. It can be assumed that $h(t', \tau)$ is a wide-sense stationary Gaussian random process in the variable t' and the responses due to different scatterers at delays τ_1 and τ_2 to be uncorrelated. This model incorporating the wide-sense stationarity and the uncorrelated scattering is widely used for characterizing time-varying systems responses [35]. Under this assumption, the power spectral density of $H(f, \tau)$, which denotes the average power output of the time-varying multipath channel as a function of time and frequency, is given by the Wiener Khinchin theorem [21].

$$\Psi(f, \tau) = \mathrm{E}\big[|\mathrm{H}(\mathrm{f}, \tau)|^2\big]. \tag{2.6}$$

The range of values of delay, τ, and the frequencies (two-sided), f, over which $\Psi(f,\tau)$ is non-zero are defined as the delay spread (T_{d}) and the Doppler spread (B_{d}) of the channel, respectively. The inverse of the delay spread is defined as the coherence bandwidth (B_{c}) and the inverse of the Doppler spread is defined as the coherence time (T_{c}) of the channel. Coherence time and coherence bandwidth denote the range of time scales and frequencies over which the variations caused due to the channel are constant. In other words, for two frequencies f_1, f_2 such that $|f_1 - f_2| \leq B_{\mathrm{c}}$, and two time instants τ_1 and τ_2 such that $|\tau_1 - \tau_2| \leq T_{\mathrm{c}}$, the power spectral density satisfies $\Psi(f_1,\tau_1) = \Psi(f_2,\tau_2)$ [36]. A finite dimensional representation of (2.5) is obtained by sampling the variables τ and f at a resolution $\Delta\tau$ and Δf such that $\tau \times f \in [0, T_{\mathrm{d}}] \times [-\frac{B_{\mathrm{d}}}{2}, \frac{B_{\mathrm{d}}}{2}]$. Specifically, each delay-Doppler grid of size $\Delta\tau \times \Delta f$ consists of all the paths whose delay and Doppler shifts fall within that grid.

$$y(t') = \sum_{q=-Q/2}^{Q/2-1} \sum_{p=0}^{P-1} H(p,q) r(t' - p\Delta\tau) e^{j2\pi q \Delta f t'} + w(t'), \tag{2.7}$$

where $Q = \lceil \frac{B_{\mathrm{d}}}{\Delta f} \rceil$ and $P = \lceil \frac{T_{\mathrm{d}}}{\Delta\tau} \rceil$. In a rich scattering like an urban environment, each of the grid points is populated by at least one path. Further, if the sampling resolutions are chosen such that $\Delta\tau > T_{\mathrm{c}}$ and $\Delta f > B_{\mathrm{c}}$, the coefficients $H(p,q)$ are independent of each other [37]. Thus, the vector $\boldsymbol{h}$ defined as $\boldsymbol{h} = \mathrm{vec}(\boldsymbol{H})$, is a multivariate Gaussian random variable with independent entries. $\boldsymbol{h}$ will be referred as the channel state vector and $\boldsymbol{H}$ as the channel state matrix from now on. For simplicity, in this chapter, we assume that the processing interval and the signal energy are chosen so that the radar can estimate the channel state vector in each tracking interval without any errors. This assumption is equivalent to considering that the channel state information is available at the receiver. In the next chapter,

we relax this assumption, and estimate the channel and the target state jointly in each tracking interval.

It is evident from the representation given in Eq. (2.7) that the received signal is a linear combination of independent time shifted (multipath) and frequency shifted (Doppler) versions of the back scattered signal from the target. Hence, the representation in Eq. (2.7) provides two kinds of additional diversity; delay diversity and the Doppler diversity. Similar representations are used in problems related to communication in fading wireless channels [37].

2.2.2 Signal Model

In this subsection, the signal model is described. The radar transmits a coherent pulse train of L pulses, with a pulse repetition interval of t_{p} seconds in each tracking interval. Pulse train signaling enables the radar receiver to estimate both the range and the Doppler without any ambiguity [38]. The corresponding transmitted signal is given as

$$s(t') = \sqrt{E}\sum_{l=0}^{L-1} a_l(t' - lt_{\mathrm{p}}),\ 0 \leq t' \leq \Delta t, \tag{2.8}$$

where $a_l(t')$ is the unit energy transmitted waveform in the l^{th} pulse and E is the signal energy per pulse and Δt is the system sampling time.

In each pulse, a spread-spectrum waveform [36] is transmitted. Apart from a few works, [39], [40], there has not been much research in the literature that use spread spectrum waveforms in the radar context. The spread-spectrum waveform $a(t')$ takes the form

$$a(t') = \sum_{g=0}^{G-1} a_g v(t' - gt_{\mathrm{c}}), \text{such that} \sum_{g=0}^{G-1} a_g a_{g-k} \approx 0, k \neq g, \tag{2.9}$$

where t_c is the chip duration, $v(t')$ is a rectangular waveform of duration t_c, G is the number of chips in each pulse, and $\{a_g, g = 0, \cdots, G-1\}$ is the spreading code corresponding to the spread-spectrum waveform $a(t')$. The bandwidth corresponding to this waveform is given by $B = \frac{1}{t_c}$.

2.2.3 State-Space Model

We consider K targets moving in a two-dimensional plane and denote the position and velocity of the k^{th} target as (x_k, y_k) and $(\dot{x}_k, \dot{y}_k)$, respectively. The target state at time t is then represented by the vector $\boldsymbol{\theta}_{k,t} = [x_{k,t}, y_{k,t}, \dot{x}_{k,t}, \dot{y}_{k,t}]^T$. We assume in this chapter that all the targets follow linear trajectories. The dynamics of the k^{th} target at time t are then described by

$$\boldsymbol{\theta}_{k,t} = \boldsymbol{F}_k \boldsymbol{\theta}_{k,t-1} + \boldsymbol{v}_{k,t-1}, \tag{2.10}$$

where $\boldsymbol{F}_k$ is the state transition matrix corresponding to the k^{th} target and is given as

$$\boldsymbol{F}_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ for } k = 1, 2, \cdots, K. \tag{2.11}$$

Here Δt is the system sampling time, which corresponds to the time interval after which the processing is done, and we refer to it as tracking interval. $\boldsymbol{v}_{k,t}$ denotes the error in the state model which is assumed to be Gaussian distributed, with a zero

mean and a covariance matrix given by [2]:

$$\boldsymbol{\Sigma}_{\mathrm{v},k} = \epsilon_k \begin{bmatrix} \frac{1}{3}\Delta t^3 & 0 & \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{3}\Delta t^3 & 0 & \frac{1}{2}\Delta t^2 \\ \frac{1}{2}\Delta t^2 & 0 & \Delta t & 0 \\ 0 & \frac{1}{2}\Delta t^2 & 0 & \Delta t \end{bmatrix}, \tag{2.12}$$

where ϵ_k is the intensity of the noise process for the k^{th} target. Using Eq. 2.10, and concatenating the state vectors of all the targets, an overall target state transition equation is obtained as

$$\boldsymbol{\theta}_t = \boldsymbol{F}\boldsymbol{\theta}_{t-1} + \boldsymbol{v}_{t-1}, \tag{2.13}$$

where

$\boldsymbol{\theta}_t : 4K \times 1$ joint target state defined as $\boldsymbol{\theta}_t = \begin{bmatrix} \boldsymbol{\theta}_{1,t}^T, \cdots, \boldsymbol{\theta}_{K,t}^T \end{bmatrix}^T$

$\boldsymbol{F} : 4K \times 4K$ matrix representing the overall state transition matrix, which is defined as $\boldsymbol{F} = \mathrm{blkdiag}\{\boldsymbol{F}_1, \cdots \boldsymbol{F}_K\}$

$\boldsymbol{v}_t : 4K \times 1$ overall additive noise given as $\boldsymbol{v}_t = \begin{bmatrix} \boldsymbol{v}_{1,t}^T, \cdots, \boldsymbol{v}_{K,t}^T \end{bmatrix}^T$ with noise covariance matrix $\boldsymbol{\Sigma}_{\mathrm{v}} = \mathrm{blkdiag}\{\boldsymbol{\Sigma}_{\mathrm{v},1}, \cdots, \boldsymbol{\Sigma}_{\mathrm{v,K}}\}$

2.2.4 Measurement Model

Using (2.1) and (2.8) in (2.7), the received signal due to a single target can be expressed as

$$\begin{aligned}\tilde{y}(t') &= \sqrt{E}\sum_{q=-Q/2}^{Q/2-1}\sum_{p=0}^{P-1}\tilde{\alpha}H(p,q)\sum_{l=0}^{L-1}a_l(t'-lt_{\mathrm{p}}-p\Delta\tau-\tau_{\mathrm{d}})e^{j2\pi\nu_{\mathrm{d}}(t'-p\Delta\tau)}e^{j2\pi q\Delta ft'}+w(t),\\ &\approx \sqrt{E}\sum_{q=-Q/2}^{Q/2-1}\sum_{p=0}^{P-1}\tilde{\alpha}H(p,q)\sum_{l=0}^{L-1}a_l(t'-lt_{\mathrm{p}}-p\Delta\tau-\tau_{\mathrm{d}})e^{j2\pi\nu_{\mathrm{d}}lt_{\mathrm{p}}}e^{j2\pi q\Delta flt_{\mathrm{p}}}+w(t').\end{aligned} \tag{2.14}$$

The term $e^{j2\pi\nu_{\mathrm{d}}t'}$ is approximated as $e^{j2\pi\nu_{\mathrm{d}}lt_{\mathrm{p}}}$ (constant) and $e^{j2\pi q\Delta ft'}$ is approximated as $e^{j2\pi q\Delta flt_{\mathrm{p}}}$ within each $a(t')$, since it is a narrowband pulse. Also the term $e^{-j2\pi\nu_{\mathrm{d}}p\Delta\tau}$ is ignored as the variation in this terms are negligible when compared to the variations in other terms. The superscript $\sim$ is used to emphasize that the signal, and the scattering coefficient correspond to a single target.

The overall received signal due to all the targets is given as

$$y(t')=\sum_{k=1}^{K}\sqrt{E}\sum_{q=-Q/2}^{Q/2-1}\sum_{p=0}^{P-1}\tilde{\alpha}_kH(p,q)\sum_{l=0}^{L-1}a_l(t'-lt_{\mathrm{p}}-p\Delta\tau-\tau_{k,\mathrm{d}})e^{j2\pi\nu k,\mathrm{d}lt_{\mathrm{p}}}e^{j2\pi q\Delta flt_{\mathrm{p}}}+w(t'), \tag{2.15}$$

where $\tau_{k,\mathrm{d}},\nu_{k,\mathrm{d}},\tilde{\alpha}_k$ represent the delay, Doppler and the RCS of the k^{th} target, respectively. The distance between the targets is smaller compared to the distance between the targets and the radar, and hence the channel state matrix is assumed to remain the same for all the targets.

2.3 Sparse Modeling

In this section, we develop a sparse measurement model for Eq. (2.15). Consider the received signal in Eq.(2.15) and discretize the delay-Doppler plane into grid points such that the delay and Doppler corresponding to each target falls within one specific grid point i.e.,

$$\begin{aligned} \tau_{k,\mathrm{d}} &= m_k^1 \Delta\tau, \\ \nu_{k,\mathrm{d}} &= m_k^2 \Delta\nu, \end{aligned} \tag{2.16}$$

for $k = 1, 2, \cdots, K$, where m_k^1 and m_k^2 represent the indices of the discretized delay and Doppler of the k^{th} target. If the delay-Doppler plane is discretized into M_1 points along the delay dimension and M_2 points along the Doppler dimension, corresponding to a total of $M = M_1 M_2$ grid points in the region of interest, the received signal for the k^{th} target can be expressed as

$$\tilde{y}_{k,t} = \sqrt{E} \sum_{q=-Q/2}^{Q/2-1} \sum_{p=0}^{P-1} \tilde{\alpha}_k H(p,q) \sum_{l=0}^{L-1} a_l(t' - lt_{\mathrm{p}} - p\Delta\tau - m_k^1\Delta\tau) e^{j2\pi(m_k^2\Delta f + q\Delta f)lt_{\mathrm{p}}} + w(t'). \tag{2.17}$$

The received signal is now sampled at a rate $f_{\mathrm{s}} = B$ so that one sample from each chip is collected. Consider N samples around a reference point[4] in each pulse. The sampling resolution of the delay-Doppler grid is chosen to commensurate with the sampling of the signal, i.e., $\Delta\tau = \frac{1}{f_{\mathrm{s}}}$ and $\Delta f = \frac{1}{\delta t}$. The corresponding discrete-time

[4]The choice of the reference point can be arbitrary. In this chapter it is chosen to be the predicted state of the first target.

signal is then given by

$$\tilde{y}_k(nt_{\rm s}) = \sqrt{E}\sum_{q=-Q/2}^{Q/2-1}\sum_{p=0}^{P-1}\tilde{\alpha}_k H(p,q)\sum_{l=0}^{L-1}a_l(nt_{\rm s}-lt_{\rm p}-pt_{\rm s}-m_k^1 t_{\rm s})e^{j2\pi(m_k^2+q)l\frac{t_{\rm p}}{\delta t}}+w(t). \tag{2.18}$$

Expressing (2.18) in a matrix form,

$$\tilde{\boldsymbol{y}}_k = \sqrt{E}\sum_{q=-Q/2}^{Q/2-1}\sum_{p=0}^{P-1}\tilde{\alpha}_k H(p,q)\underbrace{\Big(\boldsymbol{F}(q,m_k^2)\otimes\boldsymbol{J}(p,m_k^1)\Big)}_{\boldsymbol{\phi}(p,q,m_k^1,m_k^2)}\boldsymbol{s}+\boldsymbol{w}, \tag{2.19}$$

where

$\tilde{\boldsymbol{y}}_k : LN\times 1$ received signal vector corresponding to the k^{th} target

$\boldsymbol{F}(q,m_k^2) : L\times L$ Doppler modulation matrix defined as $\text{diag}\big\{1, e^{\frac{j2\pi(q+m_k^2)}{L}},\cdots,e^{\frac{j2\pi(q+m_k^2)(L-1)}{L}}\big\}$

$\boldsymbol{J}(p,m_k^1) : N\times G$ time shift matrix defined as $\begin{bmatrix}\boldsymbol{0}^T_{(m_k^1+p)\times G} & \boldsymbol{I}_G & \boldsymbol{0}^T_{(N-G-m_k^1-p)\times G}\end{bmatrix}^T$

$\boldsymbol{s}$: $LG\times 1$ column vector obtained by stacking the spreading code $\boldsymbol{a}$, L times as $[\boldsymbol{a}^T,\boldsymbol{a}^T,\cdots,\boldsymbol{a}^T]^T$.

$\boldsymbol{w}$: $LN\times 1$ complex additive white Gaussian noise with zero mean and covariance matrix $\sigma_{\rm w}^2\boldsymbol{I}_{LN}$.

In obtaining Eq. (2.19), it is assumed that all the samples of the received waveform $\tilde{y}_k(t')$ fall within the sampling window of size N. By further simplifying (2.19),

$$\tilde{\boldsymbol{y}}_k = \tilde{\alpha}_k\sqrt{E}\boldsymbol{\Phi}_k\boldsymbol{h}+\boldsymbol{w}, \tag{2.20}$$

where

$\boldsymbol{\Phi}_k : LN\times PQ$ defined as $\boldsymbol{\Phi}_k = [\boldsymbol{\phi}(1,1,m_k^1,m_k^2)\cdots,\boldsymbol{\phi}(p,q,m_k^1,m_k^2),\cdots,\boldsymbol{\phi}(P,Q,m_k^1,m_k^2)]$

$\boldsymbol{h}$: $PQ\times 1$ defined as $\text{vec}(\boldsymbol{H})$, where $\boldsymbol{H}$ is a $P\times Q$ matrix with elements $[\boldsymbol{H}]_{pq} = H(p,q)$ and $\boldsymbol{h}\sim\mathbb{C}\mathcal{N}(\boldsymbol{0},\boldsymbol{\Sigma}_{\rm h})$.

When there are K targets, each of these targets will occupy one specific grid point. The received signal due to all the targets is then given as

$$y(t') = \sqrt{E}\sum_{m=1}^{M}\alpha_m \sum_{q=-Q/2}^{Q/2-1}\sum_{p=0}^{P-1} H(p,q)\sum_{l=0}^{L-1} a_l(t-lt_{\mathrm{p}}-p\Delta\tau-m^1\Delta\tau)e^{j2\pi(q+m^2)\Delta flt_{\mathrm{p}}}+w(t'), \tag{2.21}$$

with $m^1 = (m-1) \mod M_1$, $m^2 = \lfloor \frac{m-1}{M_1} \rfloor$ and each of the K targets located at one grid point m. We collect all such grid points where the targets are located into a set $\mathcal{K}$ and define

$$\alpha_m = \begin{cases} \tilde{\alpha}_k & \text{if } m \in \mathcal{K} \\ 0 & \text{if } m \notin \mathcal{K}. \end{cases}$$

In general, the number of targets is much smaller than the number of grid points ($K << M$). This sparsity constraint allows for the development of a sparse measurement model. As before, expressing the received signal in a vector form gives

$$\boldsymbol{y} \;=\; \sum_{m=1}^{M}\alpha_m\sqrt{E}\boldsymbol{\Phi}_m\boldsymbol{h} + \boldsymbol{w}. \tag{2.22}$$

By further simplifying Eq. (2.22),

$$\boldsymbol{y} \;=\; \sqrt{E}\boldsymbol{\Phi}\boldsymbol{\zeta} + \boldsymbol{w}, \tag{2.23}$$

where

- $\boldsymbol{\Phi}$: $LN \times MPQ$ dictionary of block-sparse model defined as $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1 \cdots \boldsymbol{\Phi}_m, \cdots, \boldsymbol{\Phi}_M]$
- $\boldsymbol{\zeta}$: $MPQ \times 1$ block-sparse vector [41] defined as $\boldsymbol{\zeta} = [\alpha_1\boldsymbol{h}^T, \cdots, \alpha_M\boldsymbol{h}^T]^T = \boldsymbol{\alpha} \otimes \boldsymbol{h}$

Observe that the dictionary matrix consists of time-frequency shifted versions of the transmitted signal in its columns. The dictionary matrix satisfies a special property which will be stated as a theorem. But before that, a notation $\boldsymbol{D}_{\kappa}^{\xi}$ is introduced below.

Let $\boldsymbol{D}_1, \cdots, \boldsymbol{D}_U$ denote diagonal matrices each of order V and let $\boldsymbol{D} = \text{diag}\{\boldsymbol{D}_1, \cdots, \boldsymbol{D}_U\}$ be a block diagonal matrix. Denote by $\boldsymbol{D}_{\kappa}^{\xi}$, a matrix obtained by moving the diagonal entries of all the matrices $\boldsymbol{D}_1, \boldsymbol{D}_2 \cdots \boldsymbol{D}_U$ in $\boldsymbol{D}$ to ξ^{th} sub-diagonal below or above their respective principal diagonals, depending on whether ξ is positive or negative. Further, all the diagonal matrices in the block diagonal matrix $\boldsymbol{D}$ are moved to the κ^{th} sub-diagonal below or above the principal diagonal of $\boldsymbol{D}$, depending on whether κ is positive or negative. Thus, the superscript denotes the offset of the elements of $\boldsymbol{D}_1, \boldsymbol{D}_2 \cdots \boldsymbol{D}_U$ from their respective principal diagonals and the subscript denotes the offset of the matrices $\boldsymbol{D}_1, \boldsymbol{D}_2 \cdots \boldsymbol{D}_U$ from the principal diagonal of $\boldsymbol{D}$. An example for the notation is provided below.

Example: Let

$$\boldsymbol{D} = \begin{bmatrix} d_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & d_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & d_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & d_{21} & 0 & 0 \\ 0 & 0 & 0 & 0 & d_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & d_{23} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_2 \end{bmatrix},$$

then

$$\boldsymbol{D}_1^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ d_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & d_{12} & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{D}_1^1 & \mathbf{0} \end{bmatrix},$$

and,

$$\boldsymbol{D}_{-1}^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & d_{11} & 0 \\ 0 & 0 & 0 & 0 & 0 & d_{12} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \boldsymbol{D}_1^{-1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

with

$$\boldsymbol{D}_1^1 = \begin{bmatrix} 0 & 0 & 0 \\ d_{11} & 0 & 0 \\ 0 & d_{12} & 0 \end{bmatrix} \text{ and } \boldsymbol{D}_1^{-1} = \begin{bmatrix} 0 & d_{11} & 0 \\ 0 & 0 & d_{12} \\ 0 & 0 & 0 \end{bmatrix}.$$

Theorem 1. *If* $L \geq \frac{M_2+Q}{2}$ *and* $\sum_{g=0}^{G-1} a_g a_{g-k} = 0, k \neq g$, *then the dictionary* $\boldsymbol{\Phi}$ *satisfies*

$$\boldsymbol{\Phi}_m^H \boldsymbol{\Phi}_n = \boldsymbol{D}_{\kappa_{M_1}(n,m)}^{\xi_{M_1}(n,m)} \tag{2.24}$$

where $\xi_{M_1}(n,m) = \lfloor \frac{n-1}{M_1} \rfloor - \lfloor \frac{m-1}{M_1} \rfloor$, $\kappa_{M_1}(n,m) = \{(n-1) \mod M_1\} - \{(m-1) \mod M_1\}$, $\boldsymbol{D} = diag(L\boldsymbol{I}_P, \cdots L\boldsymbol{I}_P)$.

Proof: From the definition of $\mathbf{\Phi}$,

$$\left[\mathbf{\Phi}_m^H \mathbf{\Phi}_n\right]_{ij} = \boldsymbol{a}^H \sum_{l=0}^{L-1} e^{\frac{j2\pi l}{L}\xi_{M_1}(n,m)+\xi_P(j,i)} \left(\boldsymbol{J}\left(\kappa_{M_1}(n,m), \kappa_P(j,i)\right)\boldsymbol{a}\right). \tag{2.25}$$

1. First, consider the case $m = n$, i.e., within the same block. For this case, $\xi_{M_1}(n,m) = 0$ and $\kappa_{M_1}(n,m) = 0$. Therefore,

$$\left[\mathbf{\Phi}_m^H \mathbf{\Phi}_m\right]_{ij} = \sum_{l=0}^{L-1} e^{\frac{j2\pi l \xi_P(j,i)}{L}} \sum_{g=0}^{G-1} a_g^* a_{g-\kappa_P(j,i)}. \tag{2.26}$$

When $\xi_P(j,i) \neq 0$, i.e., $\lfloor \frac{j-1}{P} \rfloor - \lfloor \frac{i-1}{P} \rfloor \neq 0$, corresponding to all the $P \times P$ sub-matrices that are not along the principal diagonal in the $PQ \times PQ$ matrix $\mathbf{\Phi}_m^H \mathbf{\Phi}_m$, we have

$$\left[\mathbf{\Phi}_m^H \mathbf{\Phi}_m\right]_{ij} = 0. \tag{2.27}$$

When $\xi_P(j,i) = 0$, i.e., $\lfloor \frac{j-1}{P} \rfloor - \lfloor \frac{i-1}{P} \rfloor = 0$, corresponding to all the $P \times P$ sub-matrices along the principal diagonal of $\mathbf{\Phi}_m^H \mathbf{\Phi}_m$, we again consider two possibilities. First, $\kappa_P(j,i) = 0$. In this case,

$$\left[\mathbf{\Phi}_m^H \mathbf{\Phi}_m^H\right]_{ij} = L \sum_{g=0}^{G-1} a_g^* a_g = L. \tag{2.28}$$

Since $\xi_P(j,i) = 0$, this corresponds to the case when $i = j$, or equivalently all the diagonal entries of the matrix $\mathbf{\Phi}_m^H \mathbf{\Phi}_m$ are L. When $\kappa_P(j,i) \neq 0$ or equivalently, $i \neq j$,

$$\left[\mathbf{\Phi}_m^H \mathbf{\Phi}_m\right]_{ij} = L \sum_{g=0}^{G-1} a_g a_{g-(j-i)}^* = 0. \tag{2.29}$$

Hence for $m = n$, $\mathbf{\Phi}_m^H \mathbf{\Phi}_n = L\boldsymbol{I}$.

2. Next, consider the case when $m \neq n$. In this case, $\xi_{M_1}(n,m) + \xi_P(j,i) = 0$ only when $\xi_{M_1}(n,m) = -\xi_P(j,i)$ or equivalently $\xi_{M_1}(n,m) = \xi_P(i,j)$. This corresponds to all the $P \times P$ submatrices along the $\xi_{M_1}(n,m)^{th}$ diagonal. For all other i,j, since $L \geq \frac{M_2+Q}{2}$ and $\xi_{M_1}(n,m) + \xi_P(j,i) \neq 0$, we have $\left[\mathbf{\Phi}_m^H \mathbf{\Phi}_n\right]_{ij} = 0$. The constraint on L essentially eliminates the possibility of an ambiguity in the Doppler estimation. Within each submatrix along the $\xi_{M_1}(n,m)^{th}$ diagonal, $\kappa_{M_1}(n,m) + \kappa_P(j,i) = 0$, when $\kappa_{M_1}(n,m) = \kappa_P(i,j)$. These correspond to the indices along the $\kappa_{M_1}(n,m)^{th}$ diagonal within each submatrix. For all these indices, $\left[\mathbf{\Phi}_m^H \mathbf{\Phi}_n\right]_{ij} = L$. When $\kappa_{M_1}(n,m) + \kappa_P(j,i) \neq 0$, $\left[\mathbf{\Phi}_m^H \mathbf{\Phi}_n\right]_{ij} = 0$. Hence the structure.

2.4 Sparsity-Based Multiple-Target Tracking

In a sparsity-based tracking procedure, the predicted state $\tilde{\boldsymbol{\theta}}_{k,t} = [\tilde{x}_{k,t}, \tilde{y}_{k,t}, \tilde{\dot{x}}_{k,t}, \tilde{\dot{y}}_{k,t}]^T$ of each target is computed, and it is used to decide the region of interest in the received signal. The support of the block-sparse signal [5] $\boldsymbol{\zeta}$ is then computed using the measurement at time t given by Eq. (2.23). From the support of the block-sparse signal, the estimates of the delay and Doppler of each target are computed using Eq. (2.16). Next, the estimates of the target state for each target are computed using the following equations:

$$\begin{aligned} (\hat{x}_{k,t}, \hat{y}_{k,t}) &= \frac{c\hat{\tau}_{k,t,\mathrm{d}}}{2} \tilde{\boldsymbol{u}}_{k,t}, \\ (\hat{\dot{x}}_{k,t}, \hat{\dot{y}}_{k,t}) &= \frac{c\hat{\nu}_{k,t,\mathrm{d}}}{2 f_{\mathrm{c}} (\tilde{\dot{\boldsymbol{u}}}_{k,t})^T \tilde{\boldsymbol{u}}_{k,t}} \tilde{\dot{\boldsymbol{u}}}_{k,t}, \end{aligned} \tag{2.30}$$

[5] The support of a block sparse vector is defined as the set of blocks which have non-zero norm, i.e., bsupp($\boldsymbol{a}$) as bsupp($\boldsymbol{a}$) = $\{l \;\big|\; \|\boldsymbol{a}[l]\|_{\ell_2} \neq 0\}$, where $\boldsymbol{a}[l]$ denotes the l^{th} block of elements of the vector $\boldsymbol{a}$.

where $\hat{\tau}_{k,t,\mathrm{d}}$ and $\hat{\nu}_{k,t\mathrm{d}}$ are the estimates of delay and Doppler of k^{th} target at time t, respectively, and $\tilde{\boldsymbol{u}}_{k,t}$ and $\tilde{\dot{\boldsymbol{u}}}_{k,t}$ are the unit vectors in the direction of the position and velocity of k^{th} target at time t, respectively, given by

$$\begin{aligned}\tilde{\boldsymbol{u}}_{k,t} &= \left(\frac{\tilde{x}_{k,t}}{\sqrt{\tilde{x}_{k,t}^2+\tilde{y}_{k,t}^2}}, \frac{\tilde{y}_{k,t}}{\sqrt{\tilde{x}_{k,t}^2+\tilde{y}_{k,t}^2}}\right),\\ \tilde{\dot{\boldsymbol{u}}}_{k,t} &= \left(\frac{\tilde{\dot{x}}_{k,t}}{\sqrt{\tilde{\dot{x}}_{k,t}^2+\tilde{\dot{y}}_{k,t}^2}}, \frac{\tilde{\dot{y}}_{k,t}}{\sqrt{\tilde{\dot{x}}_{k,t}^2+\tilde{\dot{y}}_{k,t}^2}}\right).\end{aligned} \tag{2.31}$$

Using this procedure, the problem of multiple-target tracking boils down to the problem of estimating the support of the block-sparse signal $\boldsymbol{\zeta}$. The support of $\boldsymbol{\zeta}$ can be found by reconstructing the sparse signal $\boldsymbol{\zeta}$ using one of the following standard sparse signal reconstruction techniques, followed by thresholding.

2.4.1 Standard Sparse Signal Reconstruction Techniques

In its most canonical form, a sparse-model consists of an $N \times 1$ measurement vector of the form $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{w}$, where $\boldsymbol{\Phi}$ is an $N \times M$ dictionary of basis vectors and $\boldsymbol{x}$ is an $M \times 1$ sparse vector, with sparsity level K, $K \ll M$, and $\boldsymbol{w}$ is the additive noise. Several algorithms have been proposed in the literature to reconstruct the sparse signal $\boldsymbol{x}$, all of which can be broadly classified into three main categories. In the absence of noise, the signal $\boldsymbol{x}$ can be recovered by solving

$$\mathrm{P}_0: \quad \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_{\ell_0} \text{ subject to } \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}.$$

Unfortunately, P_0 is a non-convex optimization problem and it is NP hard to solve [42]. A convex relaxation to P_0 is obtained by replacing the ℓ_0 norm with the ℓ_1 norm or ℓ_p

norm in general. Solutions thus obtained fall under the category of convex relaxation. Basis pursuit (BP) [43] solves the problem

$$\text{BP}: \quad \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_{\ell_1} \text{ subject to } \boldsymbol{y} = \boldsymbol{\Phi x} \ .$$

In the presence of noise, a variation of basis pursuit called as the basis pursuit denoising (BPDN) [43], [44] is used. BPDN solves

$$\text{BPDN}: \quad \arg\min_{x} \|x\|_{\ell_1} \text{ subject to } \|\boldsymbol{y} - \boldsymbol{\Phi x}\|_{\ell_2} \leq \epsilon \ ,$$

where $\epsilon > 0$ is the tuning parameter chosen based on the noise level. Dantzig selector (DS) [45], another convex relaxation optimization introduced recently, solves the optimization problem

$$\text{DS}: \quad \arg\min_{x} \|x\|_{\ell_1} \text{ subject to } \|\boldsymbol{\Phi}^H (\boldsymbol{y} - \boldsymbol{\Phi x}) \|_{\ell_\infty} \leq \mu \ ,$$

where $\mu > 0$ is the tuning parameter. When the standard deviation of the additive noise is known, and the columns of the dictionary are normalized, the parameters ϵ and μ are chosen to be $\sqrt{2 \log N}\sigma$ [43]. A second category of reconstruction methods perform a greedy iterative search for the solutions of P_0 by making locally optimal choices. Orthogonal matching pursuit (OMP) [46], [47], compressive sampling matching pursuit (CoSaMP) [48] fall under this category. The algorithm terminates when the squared error of the estimate between the consecutive iterations is below a predetermined threshold. The third category of solutions enforce a sparsity based prior on the signal $\boldsymbol{x}$ and recover the signal $\boldsymbol{x}$ by solving for the MAP estimate. Most common methods in this category are Bayesian compressive sensing (BCS) [49] and sparse Bayesian learning (SBL) [50]. The choice of the sparsity enforcing priors is problem

dependent and there are no universal priors which guarantee good performance for all the models.

Using these standard sparse signal reconstruction algorithms to reconstruct the target scene in a multipath is often computationally expensive as they involve high dimensional vectors and matrices. Hence, these reconstruction algorithms are not well suited for target tracking, which requires real time processing. Further, all these algorithms depend heavily on the choice of the tuning parameters. In Sec. 2.4.3, a projection based (PB) support recovery algorithm which exploits the structural property of the dictionary is described. PB support recovery algorithm is a simple single-step procedure that does not require convex optimization or an iterative greedy search. It does not depend on the tuning parameters and it can be applied in scenarios where the selection of tuning parameters is difficult.

2.4.2 Effect of Multipath Environment

In this section, the effect of the time-varying multipath channel and the signaling on the performance of the support recovery algorithm is analyzed. As shown in Section 2.4, multiple-target tracking problem is equivalent to a support recovery problem. In particular, the set $\mathcal{K}$ of indices that correspond to the non-zero blocks of the block sparse vector $\boldsymbol{\zeta}$ in Eq. (2.23) are to be estimated. For a known block size, there are $S = \binom{M}{K}$ possible locations where the blocks of non-zero entries can be placed in $\boldsymbol{\zeta}$. Let $\mathcal{S}$ denote the set containing all these possible locations of the non-zero blocks. Given that the actual set, $\mathcal{K}$, corresponding to the block support of $\boldsymbol{\zeta}$, is located at an index m in this set $\mathcal{S}$, an error in the estimate of the delay or the Doppler of at least one target occurs whenever the index m is associated to some other index n. To study

the performance of such a scheme, the total error probability of a multiple hypothesis testing problem, where each hypothesis corresponds to a candidate support, is used as a performance metric [51].

$$\begin{cases} \mathcal{H}_0: & \text{bsupp}(\tilde{\boldsymbol{\zeta}}) = \mathcal{S}_0, \\ \vdots & \qquad\qquad . \\ \mathcal{H}_S: & \text{bsupp}(\tilde{\boldsymbol{\zeta}}) = \mathcal{S}_S \end{cases} \tag{2.32}$$

Evaluating the total error probability for an arbitrary reconstruction method is, in general, not feasible. Hence, an upper bound on total error probability for an optimal maximum-likelihood based decision criterion is used. If all the hypothesis are assumed to be equally likely, the optimal decision rule is given as [52]

$$m^* = \arg\max_m p(\boldsymbol{y}|\mathcal{H}_m). \tag{2.33}$$

The following theorem upper bounds the error probability of wrongly identifying the support.

Theorem 2. *For the sparse signal model defined in (2.22), the total error probability,* P_K^M*, is upper bounded by*

$$\mathrm{P}_K^M \leq \frac{1}{2S} \sum_{m=1}^{S} \sum_{\substack{n=1 \\ n\neq m}}^{S} \prod_{\rho=1}^{PQ} \frac{1}{1+\lambda_\rho(m,n)}, \tag{2.34}$$

where $\lambda_1(m,n) \cdots \lambda_{PQ}(m,n)$ *are the eigenvalues of the* $PQ \times PQ$ *matrix* $\boldsymbol{\Lambda}(m,n)$ *defined as* $\boldsymbol{\Lambda}(m,n) = \frac{\mathrm{SNR}}{2L} \boldsymbol{\Sigma}_{\boldsymbol{\zeta}} \Delta\boldsymbol{\Phi}_{m,n}^H \Delta\boldsymbol{\Phi}_{m,n}$, $\mathrm{SNR} = \frac{\mathrm{EL}}{\sigma^2}$ *is the signal to noise ratio,* σ *is the standard deviation of the additive white Gaussian noise* $\boldsymbol{w}$*,* $\boldsymbol{\Sigma}_\zeta = diag\{\alpha^1, \cdots, \alpha^K\} \otimes \boldsymbol{\Sigma}_{\mathrm{h}}$ *and* $\Delta\boldsymbol{\Phi}_{m,n} = \boldsymbol{\Phi}_n - \boldsymbol{\Phi}_m$*.*

Proof. For the system model described by Eq. (2.22), first consider a binary case where there are only two supports, indexed by m and n. Denote the probability of wrongly associating the support $\mathcal{S}_m$ to $\mathcal{S}_n$ as $P_{\mathrm{c}}(m \to n|m)$. This probability is evaluated for an optimal maximum likelihood based hypothesis test. For the binary case, given that $\mathrm{bsupp}(\tilde{\boldsymbol{\zeta}}) = \mathcal{S}_m$, Eq. (2.22) can be written as

$$\boldsymbol{y} = \sqrt{E}\boldsymbol{\Phi}_m\boldsymbol{\zeta} + \boldsymbol{w}, \tag{2.35}$$

where $\boldsymbol{\Phi}_m$ is obtained by concatenating K blocks each of size $N \times PQ$ that correspond to the $\mathrm{bsupp}(\tilde{\boldsymbol{\zeta}}) = \mathcal{S}_m$ and $\boldsymbol{\zeta}$ is obtained by removing the zeros in the block-sparse vector $\tilde{\boldsymbol{\zeta}}$. Using the optimal decision rule given by Eq. (2.33), an error in wrongly associating support is given as

$$P_{\mathrm{c}}(m \to n|m) = \Pr\big\{P(\boldsymbol{y}|\mathrm{bsupp}(\tilde{\boldsymbol{\zeta}}) = \mathcal{S}_n) > P(\boldsymbol{y}|\mathrm{bsupp}(\tilde{\boldsymbol{\zeta}}) = \mathcal{S}_m)|m\big\}. \tag{2.36}$$

The subscript c emphasizes that the error probability is conditional on $\boldsymbol{\zeta}$. Under the hypothesis $\mathcal{H}_m : \mathrm{bsupp}(\tilde{\boldsymbol{\zeta}}) = \mathcal{S}_m$, $\boldsymbol{y} \sim \mathbb{C}\mathcal{N}(\sqrt{E}\boldsymbol{\Phi}_m\boldsymbol{\zeta}, \boldsymbol{\Sigma})$. Thus the conditional error probability given $\boldsymbol{\zeta}$ is given as

$$\begin{aligned} P_{\mathrm{c}}(m \to n|m) &= \Pr\big\{e^{-(\boldsymbol{y}-\sqrt{E}\boldsymbol{\Phi}_n\boldsymbol{\zeta})\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\sqrt{E}\boldsymbol{\Phi}_n\boldsymbol{\zeta})} \geq e^{-(\boldsymbol{y}-\sqrt{E}\boldsymbol{\Phi}_m\boldsymbol{\zeta})\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\sqrt{E}\boldsymbol{\Phi}_m\boldsymbol{\zeta})}|m\big\}, \\ &= \Pr\Big\{\sqrt{E}\boldsymbol{y}^H\boldsymbol{\Sigma}^{-1}\Delta\boldsymbol{\Phi}_{m,n}\boldsymbol{\zeta} + \sqrt{E}\boldsymbol{\zeta}^H\Delta\boldsymbol{\Phi}_{m,n}^H\boldsymbol{\Sigma}^{-1}\boldsymbol{y} \geq \\ &\qquad E\boldsymbol{\zeta}^H\boldsymbol{\Phi}_m^H\Sigma^{-1}\boldsymbol{\Phi}_m\boldsymbol{\zeta} - E\boldsymbol{\zeta}^H\boldsymbol{\Phi}_n^H\Sigma^{-1}\boldsymbol{\Phi}_i\boldsymbol{\zeta}|m\Big\}. \end{aligned}$$

However, $y^H\boldsymbol{\Sigma}^{-1}\Delta\boldsymbol{\Phi}_{m,n}\boldsymbol{\zeta} + \boldsymbol{\zeta}^H\Delta\boldsymbol{\Phi}_{m,n}^H\boldsymbol{\Sigma}^{-1}\boldsymbol{y}$ follows a complex normal distribution with mean $\sqrt{E}\boldsymbol{\zeta}^H\boldsymbol{\Phi}_m^H\boldsymbol{\Sigma}^{-1}\Delta\boldsymbol{\Phi}_{m,n}\boldsymbol{\zeta} + \sqrt{E}\boldsymbol{\zeta}^H\Delta\boldsymbol{\Phi}_{m,n}^H\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}_i\boldsymbol{\zeta}$ and variance-covariance matrix

$\boldsymbol{\zeta}^H \Delta\boldsymbol{\Phi}_{m,n}^H \boldsymbol{\Sigma}^{-1} \Delta\boldsymbol{\Phi}_{m,n} \boldsymbol{\zeta}$. Hence,

$$\begin{aligned} P_{\mathrm{c}}(m \to n|m) &= \Pr\Big\{ \sqrt{E}(\boldsymbol{y}^H \boldsymbol{\Sigma}^{-1} \Delta\boldsymbol{\Phi}_{m,n} \boldsymbol{\zeta} + \boldsymbol{\zeta}^H \Delta\boldsymbol{\Phi}_{m,n}^H \boldsymbol{\Sigma}^{-1} \boldsymbol{y}) \geq \\ &\quad E(\boldsymbol{\zeta}^H \boldsymbol{\Phi}_n^H \Sigma^{-1} \boldsymbol{\Phi}_n \boldsymbol{\zeta} - \boldsymbol{\zeta}^H \boldsymbol{\Phi}_m^H \Sigma^{-1} \boldsymbol{\Phi}_m \boldsymbol{\zeta}) | m \Big\}, \\ &= Q\left(\sqrt{E \boldsymbol{\zeta}^H \Delta\boldsymbol{\Phi}_{m,n}^H \boldsymbol{\Sigma}^{-1} \Delta\boldsymbol{\Phi}_{m,n} \boldsymbol{\zeta}} \right), \end{aligned} \tag{2.37}$$

where $Q(\gamma)$ is the complementary error function defined as $Q(\gamma) = \int_{x=\gamma}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$. Using the bound $Q(\gamma) \leq \frac{1}{2} e^{-\gamma^2/2}$ [53],

$$P_{\mathrm{c}}(m \to n|m) < \frac{1}{2} e^{-\frac{E \boldsymbol{\zeta}^H \Delta\boldsymbol{\Phi}_{m,n}^H \boldsymbol{\Sigma}^{-1} \Delta\boldsymbol{\Phi}_{m,n} \boldsymbol{\zeta}}{2}}. \tag{2.38}$$

The unconditional probability of error is obtained by averaging $P_{\mathrm{c}}(m \to n|m)$ over the distribution of $\boldsymbol{\zeta}$. Since $\boldsymbol{\zeta} \sim \mathbb{C}\mathcal{N}_{PQ}(0, \boldsymbol{\Sigma}_\zeta)$,

$$\begin{aligned} P(m \to n|m) &< \frac{1}{2} \int e^{-\frac{E \boldsymbol{\zeta}^H \Delta\boldsymbol{\Phi}_{m,n}^H \boldsymbol{\Sigma}^{-1} \Delta\boldsymbol{\Phi}_{m,n} \boldsymbol{\zeta}}{2}} \frac{1}{(\pi)^{PQ} |\boldsymbol{\Sigma}_\zeta|} e^{-\boldsymbol{\zeta}^H \boldsymbol{\Sigma}_\zeta \boldsymbol{\zeta}} d\boldsymbol{\zeta}, \\ &= \frac{1}{2} \frac{2|\boldsymbol{\Sigma}_\zeta^{-1}|}{|E\Delta\boldsymbol{\Phi}_{m,n}^H \boldsymbol{\Sigma}^{-1} \Delta\boldsymbol{\Phi}_{m,n} + 2\boldsymbol{\Sigma}_\zeta^{-1}|} \\ &\quad \int \frac{|E\Delta\boldsymbol{\Phi}_{m,n}^H \boldsymbol{\Sigma}^{-1} \Delta\boldsymbol{\Phi}_{m,n} + 2\boldsymbol{\Sigma}_\zeta^{-1}|}{2(\pi)^{PQ}} e^{-\boldsymbol{\zeta}^H \left(\frac{E\Delta\boldsymbol{\Phi}_{m,n}^H \boldsymbol{\Sigma}^{-1} \Delta\boldsymbol{\Phi}_{m,n} + 2\boldsymbol{\Sigma}_\zeta^{-1}}{2} \right) \boldsymbol{\varsigma}} d\boldsymbol{\zeta}, \\ &= \frac{1}{2} \frac{1}{\left| E\boldsymbol{\Sigma}_\zeta \Delta\boldsymbol{\Phi}_{m,n}^H \boldsymbol{\Sigma}^{-1} \Delta\boldsymbol{\Phi}_{m,n}/2 + \boldsymbol{I} \right|}. \end{aligned} \tag{2.39}$$

The total probability of error is now obtained by using the union bound.

$$\begin{aligned} P_{\text{err}} &= \sum_{m=1}^{S} \Pr\{\text{err}|\text{m}\}\text{P(m)}, \\ &= \frac{1}{S}\sum_{m=1}^{S} \Pr\Big\{ \bigcup_{n\neq m} P(m \to n|m)\Big\}, \\ &< \frac{1}{S}\sum_{m=1}^{S}\sum_{\substack{n=1 \\ n\neq m}}^{S} P(m \to n|m), \\ &< \frac{1}{2S}\sum_{m=1}^{S}\sum_{\substack{n=1 \\ n\neq m}}^{S} \frac{1}{|\frac{E\boldsymbol{\Sigma}_\zeta \Delta\boldsymbol{\Phi}_{m,n}^H \boldsymbol{\Sigma}^{-1}\Delta\boldsymbol{\Phi}_{m,n}}{2} + \boldsymbol{I}|}. \end{aligned} \tag{2.40}$$

Denote $\boldsymbol{\Lambda}(m,n) = \frac{E\boldsymbol{\Sigma}_\zeta \Delta\boldsymbol{\Phi}_{m,n}^H \boldsymbol{\Sigma}^{-1}\Delta\boldsymbol{\Phi}_{m,n}}{2}$ and let $\lambda_1(m,n), \lambda_2(m,n), \cdots, \lambda_{PQ}(m,n)$ be the eigenvalues of $\boldsymbol{\Lambda}(m,n)$. Then,

$$P_{\text{err}} < \frac{1}{2S}\sum_{m=1}^{S}\sum_{\substack{n=1 \\ n\neq m}}^{S}\prod_{\rho=1}^{PQ}\left(\frac{1}{1+\lambda_\rho(m,n)}\right). \tag{2.41}$$

□

Several comments are in order here. First, note that the performance metric, P_K^M, can be difficult to compute due to an exponentially large number of terms in the summation. Nevertheless, it gives various insights on how a support recovery algorithm depends on the signalling parameters and the channel. P_K^M, varies inversely with the product $PQ = T_\text{d}B_\text{d}TB$. For a given signaling scheme, i.e., for a given TB, P_K^M decreases as the product $T_\text{d}B_\text{d}$ increases. This performance advantage is due to the joint delay-Doppler diversity that is inherently present in the measurement model due to a time-varying multipath channel. By using a model which captures this joint diversity, significant performance improvement is achieved. For a given channel, i.e.,

for a given $T_{\mathrm{d}}B_{\mathrm{d}}$, P_K^M decreases as the time-bandwidth product, TB, of the signaling increases. To maximize the product PQ, and to exploit the full diversity offered by the model, a signaling scheme that maximizes the time-bandwidth product should be used. For non spread-spectrum signals, the signaling duration and the bandwidth cannot be controlled independently. As a result, by increasing the signaling duration, the diversity offered due to the Doppler spread is lost and by increasing the signaling bandwidth, the diversity offered due to the delay spread is lost. For spread-spectrum signals, signaling duration and the bandwidth can be controlled independently. In particular, the time-bandwidth product will be proportional to G which can be made arbitrarily large in principle. Hence, spread-spectrum waveforms are optimal for sparsity-based tracking in multipath scenarios.

Second, the eigenvalues λ_ρ depend on the dictionary, $\mathbf{\Phi}$, which in turn depend on the spread-spectrum sequence chosen for the transmission. This leads to a signal optimization problem. The optimal sequence to be transmitted in each tracking interval can be found by solving the following optimization problem:

$$\boldsymbol{a}^* = \arg\min_{\boldsymbol{a}} \sum_{m=1}^{S} \sum_{\substack{n=1 \\ n\neq m}}^{S} \Big(\frac{1}{1+\lambda_{\min}(\boldsymbol{a})}\Big)^{PQ}, \tag{2.42}$$

where $\lambda_{\min}(\boldsymbol{a})$ corresponds to the minimum eigenvalue of the matrix $\mathbf{\Lambda}$ defined earlier. However, this design problem is not considered in this chapter.

Finally, note also that the upper bound given in Theorem 2 may not be a tight for all SNR values. Due to the large number of terms in the summation, the bound P_K^M can be greater than one for some values of SNR, in which case the bound becomes trivial.

2.4.3 Projection-Based (PB) Support Recovery Algorithm

In this subsection, a projection-based (PB) support recovery algorithm is discussed. The PB algorithm is a single step approach for finding the support of the block-sparse vector. First, construct the dictionary $\boldsymbol{\Phi}$ for the sparse model by discretizing the delay-Doppler plane, as described in Section 2.3. Next, project the received vector onto the row space of the dictionary, and coherently combine the energy in each block of the projection vector. The projection $\boldsymbol{z}$ can be expressed as

$$\boldsymbol{z} = (\boldsymbol{I}_M \otimes \boldsymbol{h}^H)\boldsymbol{\Phi}^H \boldsymbol{y}. \tag{2.43}$$

The block support of $\boldsymbol{\zeta}$ can be then estimated by finding the indices that correspond to the maximum absolute values of the vector $\boldsymbol{z}$. Below discussion demonstrates why the algorithm works. For a noise free case,

$$\boldsymbol{z} = (\boldsymbol{I}_M \otimes \boldsymbol{h}^H)\boldsymbol{\Phi}^H \boldsymbol{y} = \sqrt{E}(\boldsymbol{I}_M \otimes \boldsymbol{h}^H)\boldsymbol{\Phi}^H \boldsymbol{\Phi}\boldsymbol{\zeta}. \tag{2.44}$$

Rewriting $\boldsymbol{\zeta}$ as $\boldsymbol{\zeta} = \boldsymbol{\alpha} \otimes \boldsymbol{h}$, and expressing the r^{th} element of $\boldsymbol{z}$ as

$$z_r = \sqrt{E}\boldsymbol{h}^H \sum_{k \in \mathcal{K}} \alpha_k \boldsymbol{\Phi}_r^H \boldsymbol{\Phi}_k \boldsymbol{h}. \tag{2.45}$$

The summation over other terms is zero as $\alpha_m = 0$ for $m \notin \mathcal{K}$.

For $r \in \mathcal{K}$, (2.45) simplifies as

$$\begin{aligned} z_r &= \alpha_r \sqrt{E}\boldsymbol{h}^H \boldsymbol{\Phi}_r^H \boldsymbol{\Phi}_r \boldsymbol{h} + \sqrt{E}\boldsymbol{h}^H \sum_{k \in \mathcal{K}-\{r\}} \alpha_k \boldsymbol{\Phi}_r^H \boldsymbol{\Phi}_k \boldsymbol{h}, \\ &= \alpha_r \sqrt{E} L \boldsymbol{h}^H \boldsymbol{h} + \sqrt{E} \sum_{k \in \mathcal{K}-\{r\}} \alpha^k \boldsymbol{h}^H \boldsymbol{D}_\kappa^\xi \boldsymbol{h}. \end{aligned} \tag{2.46}$$

The result from Theorem 1 is used to express $\boldsymbol{\Phi}_r^H \boldsymbol{\Phi}_k$ as $\boldsymbol{\Phi}_r^H \boldsymbol{\Phi}_k = \boldsymbol{D}_\kappa^\xi$, where ξ is denoted as $\xi_{M_1}(k,r)$ and κ is denoted as $\kappa_{M_1}(k,r)$ for simplicity. By expanding $\boldsymbol{D}_\kappa^\xi$, using the notation described in Section 2.3,

$$\boldsymbol{h}^H \boldsymbol{D}_\kappa^\xi \boldsymbol{h} = L \sum_{i=1}^{Q-\kappa} \sum_{j=1}^{P-\xi} h_{i+\kappa,j+\xi}^* h_{ij}. \tag{2.47}$$

Since h_{ij} are independent Gaussian random variables, they are uncorrelated. Therefore, by the law of large numbers, the summation in (2.47) is approximately zero. Hence

$$z_r = \alpha_r \sqrt{E} L \boldsymbol{h}^H \boldsymbol{h} \qquad \text{for } r \in \mathcal{K}. \tag{2.48}$$

For $r \notin \mathcal{K}$, (2.45) simplifies as

$$\begin{aligned} z_r &= \sqrt{E} \boldsymbol{h}^H \sum_{k \in \mathcal{K}} \alpha_k \boldsymbol{\Phi}_r^H \boldsymbol{\Phi}_k \boldsymbol{h}, \\ &= \sqrt{E} L \sum_{k \in \mathcal{K}} \alpha_k \sum_{i=1}^{Q+1-\kappa} \sum_{j=1}^{P+1-\xi} h_{i+\kappa,j+\xi}^* h_{ij}, \\ &\approx 0. \end{aligned} \tag{2.49}$$

Thus, in the absence of noise, the indices corresponding to the support of the block-sparse vector $\boldsymbol{\zeta}$ are exactly same as the indices that correspond to non-zero elements in the vector $\boldsymbol{z}$. When there is noise in the system, $\boldsymbol{z}$ will no longer be sparse and the locations of K maximum absolute values of $\boldsymbol{z}$ determine the support of the block-sparse vector. After finding the support of the block-sparse vector $\boldsymbol{\zeta}$, the estimates of the delay and the Doppler for each target will be computed using Eq. (2.16), and the exact position and velocity for each target will be computed using Eq. (2.30). In order to associate each non-zero index to a specific target, Eq. (2.23) is solved

for $\boldsymbol{\zeta}$, using a regular least squares approach. After $\boldsymbol{\zeta}$ is known, the estimates the target RCS, α_k for $k = 1, \cdots, K$ are computed. Using the RCS values, each target is assigned to one of the K absolute maximum values. As long as the noise is bounded such that the vector $\boldsymbol{z}$ does not have spikes at the indices that do not correspond to the exact target locations, the proposed algorithm gives a perfect target scene reconstruction. This process is repeated in each tracking interval to obtain the target state. The pseudo-code for the overall tracking method is given in Algorithm 1.

Algorithm 1 PB Support Recovery Algorithm for Target Tracking in Multipath Environment

1: **for** $t = 1 : T$ **do**
2: Compute $\tilde{\boldsymbol{\theta}}_t = \boldsymbol{F}\hat{\boldsymbol{\theta}}_{t-1}$.
3: Construct the dictionary $\boldsymbol{\Phi}_t$ using $\tilde{\boldsymbol{\theta}}_t$.
4: Construct a $M \times 1$ vector $\boldsymbol{z}_t = (\boldsymbol{I}_M \otimes \boldsymbol{h}_t^H)\boldsymbol{\Phi}_t^H \boldsymbol{y}_t$.
5: Find support $\{s_k \mid |z_{t,s_k}| > |z_{t,r}|, r \in \{1, \cdots, M\}, r \neq s_k, k = 1, \cdots, K\}$.
6: Using s_k, compute $\hat{\tau}_{k,t,\mathrm{d}}, \hat{\nu}_{k,t,\mathrm{d}}, k = 1, \cdots, K$.
7: Compute the estimate of the target state using Eqs. (2.30) and (2.31).
8: **end for**

Tracking using the proposed PB support recovery is computationally less expensive when compared to the computational complexity of other tracking methods. This computational advantage is achieved because the PB support recovery does not need a convex optimization or an iterative greedy search. Tracking using the proposed PB support recovery performs better compared to the performance of the other tracking methods. This performance advantage is achieved because the projection vector $\boldsymbol{z}$ is obtained by coherently combining the energy from all the paths. Therefore, the algorithm exploits the joint delay-Doppler diversity that is inherently present in the problem.

2.5 Numerical Results

Several numerical examples are shown in this section to demonstrate the performance and computational advantage of proposed tracking method. These examples also show the effect of the time-varying multipath modeling on the performance of the support recovery based-tracking methods.

The simulated scenario had two crossing targets ($K = 2$) moving in the region of interest. The initial position of the first target was $(1200, 900)$m and it was moving with a constant velocity of $(18, 24)$m/s; the initial position of the second target was at $(900, 1559)$m and it was moving with a constant velocity of $(15, -26)$m/s. The initial parameters were estimated before the tracking process. Such an estimation can be performed using several methods including the maximum likelihood-based estimation [54], beamforming-based estimation [55], Bayesian estimation [56], and sparsity-based estimation [32] and interested readers can refer to these works for more information. Both targets were moving along linear trajectories, and hence their state transition equations were described by Eq. (4.24) and the covariance matrices of the modeling error were given by Eq. (2.12) with $\epsilon_1 = \epsilon_2 = 4$. The tracking interval length δt was chosen to be 0.5 seconds. The RCS of the first target, $\tilde{\alpha}_1$ was 1, while that of the second target, $\tilde{\alpha}_2$ was 1.4.

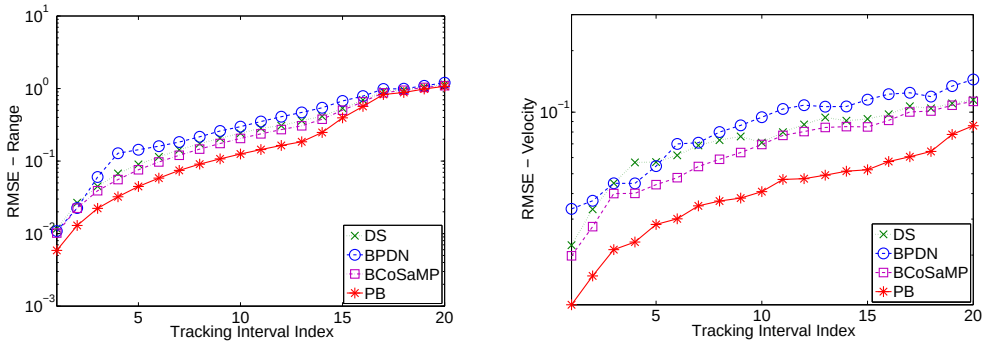
The carrier frequency, f_c, of the transmitted waveforms was 1Ghz and the bandwidth, B, of each pulse was 100Mhz. We used $L = 4$ pulses in each tracking interval. In each pulse, a spread waveform generated from a pseudo-random noise sequence of length, $G = 16$ was used. Since there was no signal optimization or pulse to pulse diversity employed, the same signal was used over all the 4 pulses and the 20 tracking intervals. In each tracking interval, the entries of the matrix $\boldsymbol{H}$ were generated independently

from a complex Gaussian distribution. The entries were scaled later such that variance of the coefficients corresponding to different delays decayed exponentially. PQ was chosen to be 6. A window of 300 samples ($N = 300$) per pulse around the predicted state of the first target in each tracking interval was considered. Such a large window size is required is accommodate the received signal vector of all the targets, when they are moving away from each other over time. The entries of $\boldsymbol{w}$ were drawn independently from a complex Gaussian distribution and were scaled to obtain the required SNR defined as

$$\text{SNR} = \frac{\text{EL}}{\sigma_{\text{w}}^2}. \tag{2.50}$$

The performance of proposed projection-based tracking method was compared with other sparsity based tracking methods that employed BPDN, DS and BCoSaMP algorithms for support recovery. The delay-Doppler plane was divided into $M_1 = 5$ and $M_2 = 5$ grid points for each target, with the grid size $\Delta\tau = 1/f_{\text{s}}$ and $\Delta f = 1/t_{\text{i}}$. This corresponded to a position and velocity estimate resolution of 1.5m and 0.3m/s, respectively in the x and the y directions. Support recovery using BPDN and DS were done by solving the optimization problems labeled BPDN and DS in Section 2.4.1, followed by thresholding. MATLAB's CVX package [57] was used to solve the convex optimization problem and the tuning parameters were chosen based on the noise variance, which was assumed to be known. Both these algorithms do not use the additional information about the block sparsity present in the problem. A BCoSaMP algorithm similar to the one used in [31] and [41] was used to solve the support recovery problem. The BCoSaMP algorithm used the additional information about the block sparsity in the problem.

All the simulations were averaged over 50 Monte Carlo iterations, and we used an SNR of 25dB.



(a) RMSE in range

(b) RMSE in velocity

Figure 2.2: Root mean-square-error for various tracking algorithms.

In Fig. 2.2, the cumulative root mean-square-error (RMSE) in the range and velocity estimates were plotted and in Fig. 2.3, actual and estimated trajectories for both the targets using different methods were plotted.

It can be seen that for all the sparsity based methods the RMSE's in the range and the velocity estimates were under 2m and 1m/s, respectively. The RMSE in the velocity estimates using PB support recovery was slightly better when compared to other methods. Further it should also me emphasized that the methods based on DS and BPDN assume that the noise variance is known for selecting the tuning parameter. The time taken by each of these algorithms is tabulated in Table 2.1.

In the second set of simulations, the performance improvement obtained due to the joint delay-Doppler diversity in the model was demonstrated. In Fig. 2.4, the upper bound on the error probability is shown against the SNR for various values of P and Q. It can be seen that as the product PQ increased, the bound on the error probability decreased significantly. This is due to additional degrees of freedom provided by the joint delay and Doppler diversity present in the urban environment. Also, for $PQ = 1$,

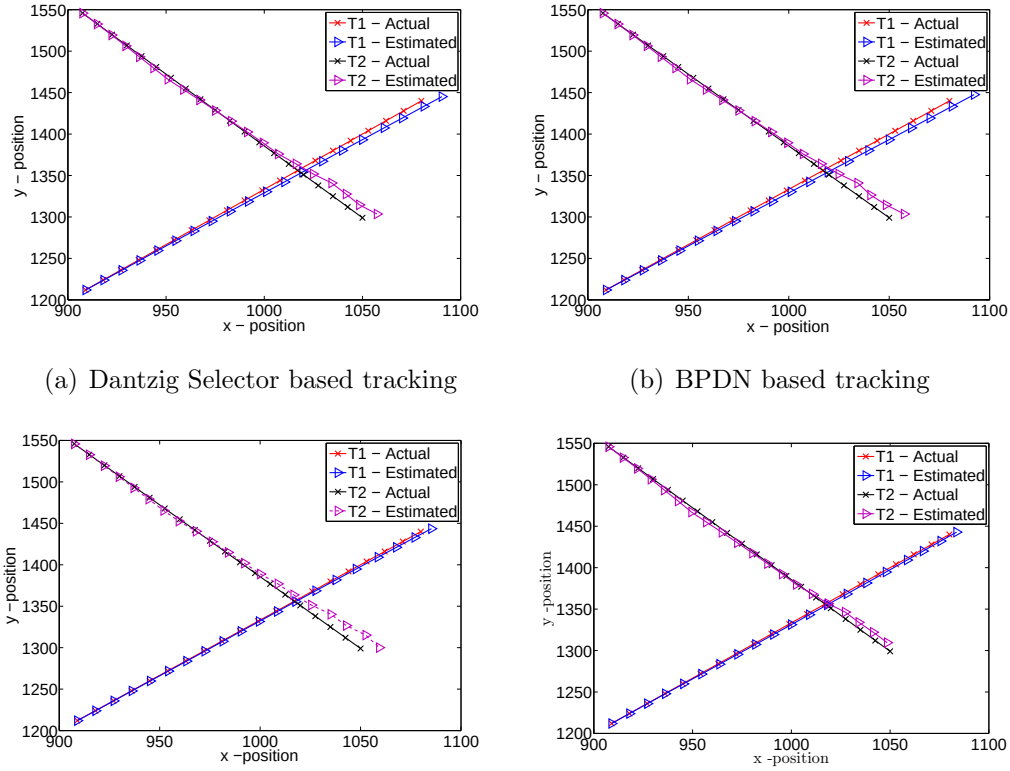
the bound was trivial and was greater than one for the range of SNR considered. To get a useful bound, the SNR has to be increased. In Fig. 3.5, the actual RMSE in the range and the velocity estimates obtained using the PB tracking for various values of P and Q. It can be seen that when the product $PQ = 1$, RMSE increased and was unbounded. For the same SNR, the performance in the presence of time-varying multipath channel model was significantly better.

Table 2.1: Average CPU time (in seconds) for various sparsity-based tracking algorithms.

	DS	**BPDN**	**BCoSaMP**	**PB**
5×5 grid	8.56	7.04	2.78	2.16
9×9 grid	94.2	19.42	10.31	6.20

2.6 Summary

In this chapter, we considered the problem of multiple-target tracking in a time-varying multipath channel. A sparse model was developed by considering a finite dimensional representation of the time-varying system function that characterizes the channel, and then discretizing the delay-Doppler plane. The target scene is then represented as a sparse signal which has non-zero values only at the indices which correspond to target locations. Tracking is then performed in two steps: predication and update. In the prediction step, a predicted value of the target state is obtained using the estimate of the target state from the previous tracking interval. This predicted state is used to construct the dictionary for the sparse model. In the update step, the estimates of the delay and Doppler of each target are obtained. Since the target



(a) Dantzig Selector based tracking

(b) BPDN based tracking

(c) BCoSaMP based tracking

(d) Projection based tracking

Figure 2.3: Actual Vs estimated trajectories using various tracking algorithms.

scene is represented as a sparse signal, problem of delay and Doppler estimation is equivalent to a sparse support recovery problem.

An upper bound on the error probability for an optimal maximum likelihood based support recovery algorithm was then derived. Using this bound, we showed that the performance of the support recovery depends inversely on the time-bandwidth product of the signaling; hence spread-spectrum waveforms are optimal for sparsity based recovery in multipath environments since their signaling duration and bandwidth can be controlled independently. With spread-spectrum signaling, the dictionary of the block-sparse model exhibits a special structure. We exploited this special structure to propose a new projection based (PB) support recovery algorithm. As the PB support

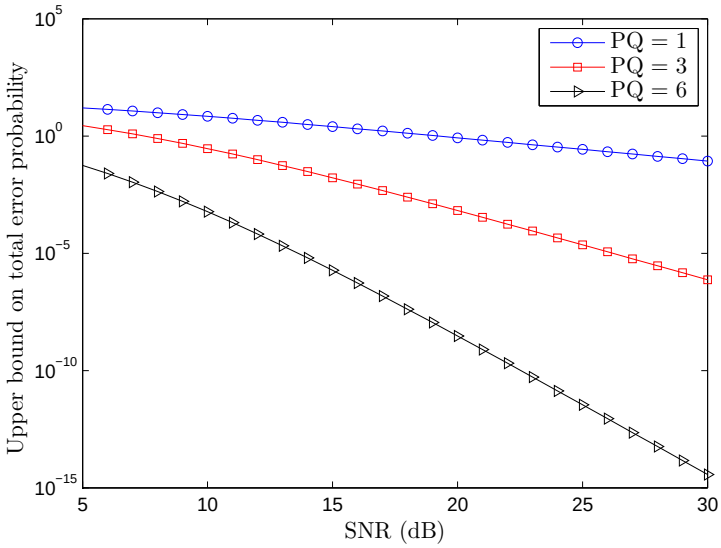
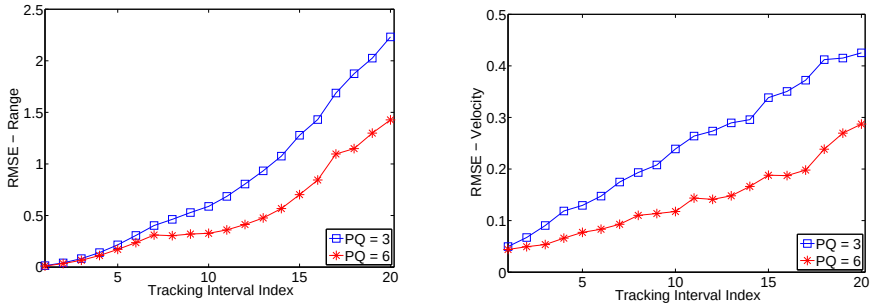


Figure 2.4: Upper bound on error probability for an optimal reconstruction.

recovery algorithm did not involve an iterative search or the need to solve a convex optimization problem, target tracking using this algorithm was computationally less intensive. Further, in the scenarios where the additive noise is bounded, PB algorithm gives exact target scene reconstruction and the corresponding tracking performance was better when compared to the performance obtained using other standard sparse signal reconstruction algorithms. Numerical simulations demonstrated that the proposed tracking algorithm takes less time when compared to the time taken by the standard sparse signal reconstruction-based tracking, and it produced lower mean-squared error (MSE) in the position and velocity estimates of the targets when compared to the MSE obtained using other methods.



(a) RMSE in range (b) RMSE in velocity

Figure 2.5: Root mean-square-error for various values of PQ.

Chapter 3

Multiple Rao Blackwellized Particle Filtering for MTT[6]

In the previous chapter, we developed a sparsity based tracking algorithm for targets moving in a time-varying multipath environment. We assumed that the channel state representing the environment was known in all the tracking intervals. In this chapter, we develop a tracking algorithm that jointly estimates the channel and the target states in each tracking interval.

3.1 Introduction

In this chapter, we use a radar network for the task of tracking multiple-targets moving in a time-varying multipath environment. As discussed in Chapter 2, such an environment offers an inherent delay-Doppler diversity which can be used to obtain

[6]Based on P. Chavali and A. Nehorai, "Scheduling and Resource Allocation in a Cognitive Radar Network for Multiple-Target Tracking", *IEEE Trans. on Signal Processing*, vol. 60, no. 2, pp. 715-729, Feb. 2012.

a better tracking performance. However, accurate priori information about the channel state that characterizes the multipath environment is required. When no prior information is available, the channel state has to be estimated along with the target state. When multiple sensors are employed, the channel state between each pair of sensors has to be estimated. Hence, the problem of MTT in a multipath environment poses a computational challenge due to the high-dimensionality of the state space.

In this chapter, we develop a particle filter for obtaining an approximate Bayesian estimate of the target and the channel state. We develop a hybrid filter, which is a combination of a multiple particle filter (MPF) and a Rao-Blackwellized particle filter (RBPF), by exploiting the structure of the state space.

Next, we derive the closed-form expressions for the PCRB on the estimates of the state vector, when the received signal at each radar is a linear combination of the delayed and Doppler-shifted versions of the signals transmitted from all the radars. Specifically, we do not assume that the signals transmitted by the individual radars are orthogonal to each other for all the delay and Doppler pairs. We then address the problems of adaptive sensor scheduling and power allocation for the radar network. Since the total costs of acquiring measurements, the communication involved with central processing, and the computational complexity of processing the measurements, increases with the number of operational radars, it is important to adaptively select a subset of operational radars and the power allocated to them at each time, to minimize the error in the estimate of the state vector. The problem of selecting a subset of sensors from a given set of possible sensors arises in various applications and has been addressed in the literature for passive networks [58]. The estimation performance is evaluated using the volume of the confidence ellipsoid as a performance metric, which is minimized for finding a suitable subset of sensors to be employed.

For an active sensor network, such as a radar network, it is also important to consider the constraints on the signal power to be transmitted, and the sensor locations while formulating the optimization problem. Few works in the past have addressed the problem of sensor scheduling for active sensor networks like a distributed MIMO radar network. In [59], [60], the authors proposed a subset selection algorithm for the task of estimating the position of a single stationary target. They do not assume the presence of multipath and assume that the signals transmitted from each radar to be orthogonal. In [61], authors considered tracking multiple targets, but they also do not consider multipath and assume that the transmitted signals are orthogonal. They perform an iterative local search to minimize the PCRB and find a subset of antennas to be employed at each time. In this chapter, we consider tracking multiple targets moving in a multipath scenario. We derive the PCRB for arbitrary transmit signals and use that as an optimization criterion for the scheduling and power allocation problems. We propose a two-pass greedy algorithm for finding a suitable antenna subset in the first pass and the power to be transmitted by the selected antennas in the second pass. Our algorithm is adaptive, and we select the antennas to be used and the power to be transmitted in each tracking interval based on the target state and the channel state estimates, which are obtained through the feedback from the receiver, with suitable constraints on the overall transmit power and communication cost.

3.2 System Model

We extend the model proposed in Chapter 2 to the case of a radar network in this chapter. Consider a network of P monostatic radars labeled as $\mathcal{P} = \{1, 2, \ldots, p, \ldots, P\}$

operating in a centralized fashion, i.e., information fusion, scheduling and resource allocation are confined to a central fusion center. The radar network is employed in the region of interest $\mathcal{R} \subset \mathbb{R}^2$, with the p^{th} radar located at (x_p, y_p). One of the P radars will act as the fusion center for the network, and, without the loss of generality, we consider the first radar to be the fusion center of the network and that it is located at (0,0). There are K point targets moving in the region of interest $\mathcal{R}$, with the position and velocity of the k^{th} target given as (x_k, y_k) and $(\dot{x}_k, \dot{y}_k)$. We abuse the notation slightly here and use the same symbols x and y to denote the x- and the y- positions of the radar and the target; we differentiate between them based on the sub-script used for indexing the radar antennas and the targets. Throughout the chapter, we use the subscript k to denote the target. All the other subscripts correspond to the radar antennas. Other than the targets, there are multiple scatters in the region $\mathcal{R}$, which can be stationary or moving at speeds comparable to the speed of the targets.

3.2.1 State-Space Model

Denote by $\boldsymbol{\theta}_{k,t}$ the state vector corresponding to the k^{th} target at time t, and let $\boldsymbol{\theta}_t = \left[\begin{array}{c} \boldsymbol{\theta}_{1,t}^T, \cdots, \boldsymbol{\theta}_{K,t}^T \end{array} \right]^T$ denote the $4K \times 1$ vector of the joint target state. Then from Eq. 2.13 of Chapter 2 we get an overall target state transition equation given as

$$\boldsymbol{\theta}_t = \boldsymbol{F}_\theta \boldsymbol{\theta}_{t-1} + \boldsymbol{v}_{\theta,t-1}. \tag{3.1}$$

The noise is assumed to be additive white Gaussian noise with covariance matrix $\boldsymbol{\Sigma}_{\mathrm{v}_\theta} = \mathrm{blkdiag}\{\boldsymbol{\Sigma}_{1,\mathrm{v}_\theta}, \cdots, \boldsymbol{\Sigma}_{K,\mathrm{v}_\theta}\}$ and $\boldsymbol{\Sigma}_{k,\mathrm{v}_\theta}$ given by Eq. 2.12.

The state transition for the channel is assumed to be a first order Markovian process, and it is described by the following equation:

$$\boldsymbol{\beta}_t = \boldsymbol{\beta}_{t-1} + \boldsymbol{v}_{\beta,t-1}, \tag{3.2}$$

where the noise is assumed to be white Gaussian, with a known covariance matrix $\boldsymbol{\Sigma}_{\mathrm{v}_\beta}$, given by

$$\boldsymbol{\Sigma}_{\mathrm{v}_\beta} = \mathrm{diag}\{\mathrm{vec}(\boldsymbol{\Sigma}_\beta)\} \otimes \boldsymbol{I}_{KN_\tau N_f}, \tag{3.3}$$

where $\boldsymbol{\Sigma}_\beta$ is a $P \times P$ matrix with $[\boldsymbol{\Sigma}_\beta]_{pq}$ denoting the variance of the mutipath channel between the p^{th} radar and q^{th} radar, and $\mathrm{diag}\{\mathrm{vec}(\boldsymbol{\Sigma}_\beta)\}$ is a $P^2 \times P^2$ diagonal matrix with $\mathrm{vec}(\boldsymbol{\Sigma}_\beta)$ along the principal diagonal. We form an extended state vector by concatenating the target state vector and the channel state vector into a single vector of dimension $4M+PMPN_\tau N_f$ defined as $\boldsymbol{\xi} = [\boldsymbol{\theta}^T, \boldsymbol{\beta}^T]^T$. The state transition equation for $\boldsymbol{\xi}$ is given as

$$\boldsymbol{\xi}_t = \boldsymbol{F}\boldsymbol{\xi}_{t-1} + \boldsymbol{v}_{\xi,t-1}, \tag{3.4}$$

where the overall state transition matrix is given as

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}_\theta & \boldsymbol{0}_{4K \times PKPN_\tau N_f} \\ \boldsymbol{0}_{PKPN_\tau N_f \times 4K} & \boldsymbol{I}_{PKPN_\tau N_f} \end{bmatrix},$$

and $\boldsymbol{v}_{\xi,t-1}$ is the additive white Gaussian noise with covariance matrix $\boldsymbol{\Sigma}_{\mathrm{v}_\xi}$. Henceforth in this chapter, when we say state vector, we refer to the extended state vector formed by concatenating the target state and the channel state.

3.2.2 Measurement Model

At each antenna, we transmit a coherent train of multiple pulses with a pulse repetition period of t_{p} seconds. The transmitted signal at the q^{th} radar is given as

$$s_q(t') = \sum_{l=0}^{L-1} a_{lq}(t' - lt_{\mathrm{p}}), \tag{3.5}$$

where $a_{lq}(t')$ is the transmitted signal in the l^{th} pulse from the q^{th} radar, and $s_q(t')$ is a unit energy signal.

Let τ_{pqm} be the total time taken for the signal to travel from the q^{th} radar to the k^{th} target and back to the p^{th} radar, and ν_{pqk} be the Doppler frequency shift due the q^{th}, p^{th} transmit-receive pair and the k^{th} target. As before, the propagation path consists of a forward transmission channel, the target itself, and a reverse transmission channel (see Fig. 2.1). The parameters τ_{pqk} and ν_{pqk} depend on the position and velocity of the k^{th} target and the positions of the p^{th} radar and the q^{th} radar. We have

$$\tau_{pqk} = \frac{1}{c}\Big\{R_{qk} + R_{pk}\Big\}, \tag{3.6}$$

and

$$\nu_{pqk} = \frac{f_{\mathrm{c}}}{c}\Big\{\dot{R}_{qk} + \dot{R}_{pk}\Big\}, \tag{3.7}$$

where c is the speed of propagation, f_{c} is the carrier frequency, R_{qk} is the range from the radar q to the target k, R_{pk} is the range from the radar p to the target k, and

$\dot{R}_{qk}$ and $\dot{R}_{pk}$ are the corresponding range rates, i.e.,

$$\begin{aligned} R_{qk} &= \sqrt{(x_q - x_k)^2 + (y_q - y_k)^2}, \\ R_{pk} &= \sqrt{(x_p - x_k)^2 + (y_p - y_k)^2}, \\ \dot{R}_{qk} &= \frac{\dot{x}_k(x_q - x_k) + \dot{y}_k(y_q - y_k)}{R_{qk}}, \text{ and} \\ \dot{R}_{pk} &= \frac{\dot{x}_k(x_p - x_k) + \dot{y}_k(y_p - y_k)}{R_{pk}}. \end{aligned} \tag{3.8}$$

Following Eq. 2.14 from Chapter 2, the received signal at the p^{th} radar, due to the signal transmitted from the q^{th} radar and bouncing off the k^{th} target is given as

$$\begin{aligned} & y_{pqk}(t') = \\ & \sqrt{\gamma_q} \sum_{n_f=-\frac{N_f}{2}}^{\frac{N_f}{2}-1} \sum_{n_\tau=0}^{N_\tau-1} \alpha_k H_{pqk}(n_f\Delta f, n_\tau\Delta\tau) s_q(t' - \tau_{pqk} - n_\tau\Delta\tau) e^{j2\pi\nu_{pqk}(t'-n_\tau\Delta\tau)} e^{j2\pi n_f\Delta f t} + w_p(t'), \\ = & \sqrt{\gamma_q} \sum_{n_f=-N_f/2}^{N_f/2-1} \sum_{n_\tau=0}^{N_\tau-1} \beta_{pqk}^{n_f n_\tau} s_q(t' - \tau_{pqk} - n_\tau\Delta\tau) e^{j2\pi(\nu_{pqk}+n_f\Delta f)t'} + w_p(t'), \\ = & \sqrt{\gamma_q} \sum_{n_f=-N_f/2}^{N_f/2-1} \sum_{n_\tau=0}^{N_\tau-1} \sum_{l=0}^{L-1} \beta_{pqk}^{n_f n_\tau} a_{lq}(t' - lt_{\mathrm{p}} - \tau_{pqm} - n_\tau\Delta\tau) e^{j2\pi(\nu_{pqk}+n_f\Delta f)lt_{\mathrm{p}}} + w_p(t'), \end{aligned} \tag{3.9}$$

where

- α_k is the radar cross section (RCS) of the k^{th} target
- γ_q is the energy of the signal transmitted from the q^{th} radar
- $N_f = \lceil \frac{B_{\mathrm{d}}}{\Delta f} \rceil$ and $N_\tau = \lceil \frac{T_{\mathrm{d}}}{\Delta\tau} \rceil$ represent the number of the different delay and the Doppler paths,respectively
- $\beta_{pqk}^{n_f n_\tau} = \alpha_k H_{pqk}(n_f\Delta f, n_\tau\Delta\tau)$ and

- $w_p(t')$ is the additive noise at the p^{th} receiver which is assumed to be circularly symmetric, complex, white, and following a Gaussian distribution.

The resolution of the sampling of the delay-Doppler plane is chosen to match the signaling duration and bandwidth, i.e., $\Delta\tau = \frac{1}{B}$ and $\Delta f = \frac{1}{T}$. We now sample the received signal at a rate $f_{\rm s} = B$, and consider N samples around a reference point (obtained by using the predicted state of the first target) in each pulse repetition interval. The corresponding discrete-time signal is then given by

$$y_{pqk}(nt_{\rm s}) = \sqrt{\gamma_q} \sum_{n_f=-N_f/2}^{N_f/2-1} \sum_{n_\tau=0}^{N_\tau-1} \sum_{l=0}^{L-1} \beta_{pqk}^{n_f n_\tau} a_{lq}(nt_{\rm s} - lt_{\rm p} - \tilde{\tau}_{pqk} - n_\tau t_{\rm s}) e^{j2\pi(\nu_{pqk}+n_f\Delta f)lt_{\rm p}} + w_p(nt_{\rm s}). \tag{3.10}$$

Here $\tilde{\tau}_{pqk} = f_{\rm s}\tau_{pqk}$ is the delay in the discrete domain. Expressing Eq. (3.10) in a matrix form, we get

$$\boldsymbol{y}_{pqk} = \sqrt{\gamma_q} \sum_{n_f=-N_f/2}^{N_f/2-1} \sum_{n_\tau=0}^{N_\tau-1} \beta_{pqk}^{n_f n_\tau} \underbrace{\Big(\boldsymbol{\Upsilon}(n_f) \otimes \boldsymbol{\Gamma}(n_\tau)\Big)\boldsymbol{s}_q}_{\boldsymbol{\phi}(n_\tau, n_f)} + \boldsymbol{w}_p, \tag{3.11}$$

where

- $\boldsymbol{y}_{pqk}$ is a $LN \times 1$ received signal vector at the p^{th} antenna due to the signal transmitted from the q^{th} antenna and bouncing off the m^{th} target
- $\boldsymbol{\Upsilon}(n_f)$ is a $L \times L$ Doppler modulation matrix defined as $\text{diag}\big\{1, e^{j2\pi(\nu_{pqk}+n_f\Delta f)t_{\rm p}}, \cdots, e^{j2\pi(\nu_{pqk}+n_f\Delta f)(L-1)t_{\rm p}}\big\}$
- $\boldsymbol{\Gamma}(n_\tau)$ is a $N \times G$ time shift matrix defined as $\begin{bmatrix} \boldsymbol{0}_{(\tilde{\tau}_{pqk}+n_\tau)\times G} & \boldsymbol{I}_G & \boldsymbol{0}_{(N-G-\tilde{\tau}_{pqk}-n_\tau)\times G} \end{bmatrix}^T$

- $\boldsymbol{s}_q$ is a $LG \times 1$ column vector obtained by stacking the transmitted signal in each pulse from the q^{th} antenna, i.e., $\boldsymbol{s}_q = [\boldsymbol{a}_{0q}^T, \boldsymbol{a}_{1q}^T, \ldots, \boldsymbol{a}_{L-1q}^T]^T, \boldsymbol{a}_{lq} = [a_{lq}^0, \ldots, a_{lq}^{G-1}]^T$
- $\boldsymbol{w}_p$ is a $LN \times 1$ complex additive white Gaussian noise at the p^{th} receiver with zero mean and covariance matrix $\boldsymbol{\Sigma}_{\mathrm{w},p} = \sigma_{\mathrm{w},p}^2 \boldsymbol{I}_{LN}$

In obtaining Eq. (3.11), we assumed that all the samples of the received waveform $y_{pqk}(t)$ fall within the sampling window of size N and that the pulse width is greater than t_{s} seconds. The second assumption ensures that there is at least one sample from each pulse. By further simplifying Eq. (3.11), we get

$$\boldsymbol{y}_{pqk} = \sqrt{\gamma_q}\boldsymbol{\Phi}_{pqk}\boldsymbol{\beta}_{pqk} + \boldsymbol{w}_p, \tag{3.12}$$

where

- $\boldsymbol{\Phi}_{pqk}$ is a $LN \times N_\tau N_f$ matrix defined as $\boldsymbol{\Phi}_{pqk} = [\cdots, \boldsymbol{\phi}(n_\tau, n_f), \cdots]$ and
- $\boldsymbol{\beta}_{pqk}$ is a $N_\tau N_f \times 1$ vector defined as $[\cdots, \beta_{pqk}^{n_\tau n_f}, \cdots]^T$

The received signal at the p^{th} antenna due to all the targets and all the antennas is then given as

$$\boldsymbol{y}_p = \sum_{q=1}^{P}\sum_{k=1}^{K} \sqrt{\gamma_q}\boldsymbol{\Phi}_{pqk}\boldsymbol{\beta}_{pqk} + \boldsymbol{w}_p. \tag{3.13}$$

The final measurement equation is obtained by concatenating the measurement vectors at all the antennas and is given as

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\beta} + \boldsymbol{w}, \tag{3.14}$$

where

- $\boldsymbol{y} = [\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_P^T]^T$ is a $LNP \times 1$ vector of the received signal
- $\boldsymbol{\Phi} = \text{blkdiag}\{\boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_p, \ldots, \boldsymbol{\Phi}_P\}$ is a $LNP \times PKPN_\tau N_f$ matrix, where $\boldsymbol{\Phi}_p = \{\sqrt{\gamma_1}\boldsymbol{\Phi}_{p11}, \ldots, \sqrt{\gamma_P}\boldsymbol{\Phi}_{pPK}\}$
- $\boldsymbol{\beta} = [\boldsymbol{\beta}_{111}^T, \ldots, \boldsymbol{\beta}_{PPK}^T]^T$ is a $PKPN_\tau N_f \times 1$ vector of the channel state and
- $\boldsymbol{w} = [\boldsymbol{w}_1^T, \ldots, \boldsymbol{w}_P^T]^T$ is a $LNP \times 1$ measurement noise vector with covariance matrix $\boldsymbol{\Sigma}_\text{w} = \text{blkdiag}\{\boldsymbol{\Sigma}_{\text{w},1}, \ldots, \boldsymbol{\Sigma}_{\text{w},P}\}$.

3.3 Multiple Rao-Blackwellized Particle Filter

In this section, we propose a new hybrid filter that partitions the state space into lower dimensional subspaces and generates the particles from the lower dimensional state space. Our method is based on the combination of a multiple particle filter [62] and a Rao-Blackwellized particle filter [63], [64], [65]. The idea behind a multiple particle filter is to partition the state space into subspaces of lower dimension, such that the state transition of each subspace is independent of other subspaces, and then to employ multiple particle filters operating on each subspace independently. The idea behind a Rao-Blackwellized particle filter is to partition the state space, such that conditioned on one partition the system becomes linear and Gaussian. This partition can then be marginalized out analytically using a Kalman filter. The Rao-Blackwell theorem states that the variance of the estimates obtained after Rao-Blackwellization is less than the variance of the original estimate. We use these ideas to develop a hybrid filter which is a combination of both MPF and a RBPF. We first partition the

state space as target state and channel state, i.e., $\boldsymbol{\xi} = [\boldsymbol{\theta}^T, \boldsymbol{\beta}^T]^T$. The joint posterior distribution at time t, given the measurements up to t can be expressed as

$$p(\boldsymbol{\xi}_t|\boldsymbol{y}_{1:t}) \;\; = \;\; p(\boldsymbol{\beta}_t|\boldsymbol{\theta}_t, y_{1:t})p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t}) \tag{3.15}$$

Given a particle $\boldsymbol{\theta}_t^{(i)}$, the measurement model given in Eq. (3.14), is linear and Gaussian in the channel state vector $\boldsymbol{\beta}$. Hence, we use a Kalman filter to obtain the measurement and time updates corresponding to the partition $\boldsymbol{\beta}$. Next, we further partition the target state into smaller subspaces where each partition corresponds to the state of a single target. Since the state transition corresponding to each target is independent of other targets, the distribution $p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t})$ can be expressed as

$$p(\boldsymbol{\theta}_t|\boldsymbol{y}_{1:t}) = \prod_{k=1}^{K} p(\boldsymbol{\theta}_{k,t}|\boldsymbol{y}_{1:t}), \tag{3.16}$$

We employ one particle filter for each partition, and approximate the distributions $p(\boldsymbol{\theta}_{k,t}|\boldsymbol{y}_{1:t}), k = 1, \cdots, K$ using random measures defined by $(\boldsymbol{\theta}_{k,t}^{(i)}, w_{k,t}^{(i)})$. The corresponding weight update equations can be expressed as [62]

$$\tilde{w}_{k,t}^{(i)} \propto w_{k,t-1}^{(i)} p(\boldsymbol{y}_t|\boldsymbol{\theta}_{k,t}^{(i)}, \boldsymbol{y}_{1:t-1}). \tag{3.17}$$

The density $p(\boldsymbol{y}_t|\boldsymbol{\theta}_{k,t}^{(i)}, \boldsymbol{y}_{1:t-1}) \neq p(\boldsymbol{y}_t|\boldsymbol{\theta}_{k,t}^{(i)})$. This is because the measurements $\boldsymbol{y}_{1:t-1}$ contain the information about the channel state $\boldsymbol{\beta}_t$ and the target state $\boldsymbol{\theta}_{-k,t} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{k-1}, \boldsymbol{\theta}_{k+1}, \ldots, \boldsymbol{\theta}_K]_t^T$. We express the distribution $p(\boldsymbol{y}_t|\boldsymbol{\theta}_{k,t}^{(i)}, \boldsymbol{y}_{1:t-1})$ as

$$p(\boldsymbol{y}_t|\boldsymbol{\theta}_{k,t}^{(i)}, \boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{y}_t|\boldsymbol{\theta}_{k,t}^{(i)}, \boldsymbol{\theta}_{-k,t}, \boldsymbol{\beta}_t) \times p(\boldsymbol{\beta}_t|\boldsymbol{\theta}_{k,t}^{(i)}, \boldsymbol{y}_{1:t-1}) \times p(\boldsymbol{\theta}_{-k,t}|\boldsymbol{\theta}_{k,t}^{(i)}, \boldsymbol{y}_{1:t-1}) \mathrm{d}\boldsymbol{\beta}_t \mathrm{d}\boldsymbol{\theta}_{-k,t}, \tag{3.18}$$

and the distributions $p(\boldsymbol{\beta}_t|\boldsymbol{\theta}^{(i)}_{k,1:t}, \boldsymbol{y}_{1:t-1})$ and $p(\boldsymbol{\theta}_{-k,t}|\boldsymbol{\theta}^{(i)}_{k,t}, \boldsymbol{y}_{1:t-1})$ as

$$p(\boldsymbol{\beta}_t|\boldsymbol{\theta}^{(i)}_{k,t}, \boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{\beta}_t|\boldsymbol{\beta}_{t-1}) p(\boldsymbol{\beta}_{t-1}|\boldsymbol{y}_{1:t-1}) \mathrm{d}\boldsymbol{\beta}_{t-1}, \tag{3.19}$$

and

$$p(\boldsymbol{\theta}_{-k,t}|\boldsymbol{\theta}^{(i)}_{k,t}, \boldsymbol{y}_{1:t-1}) = \prod_{\substack{n=1 \\ n\neq k}}^{K} \int p(\boldsymbol{\theta}_{n,t}|\boldsymbol{\theta}_{n,t-1}) p(\boldsymbol{\theta}_{n,t-1}|\boldsymbol{y}_{1:t-1}) \mathrm{d}\boldsymbol{\theta}_{n,t-1}. \tag{3.20}$$

We identify Eqs. (3.19) and (3.20) to be the time update equations corresponding to the Kalman filter and the $K-1$ particle filters, respectively. Also, Eq. (3.19) can be computed analytically since the distributions defined in this equation are Gaussian [64]. Following the similar procedure, we compute the weight update equations corresponding to all K particle filters. Finally, we compute the marginal $p(\boldsymbol{\beta}_t|y_{1:t})$ using

$$p(\boldsymbol{\beta}_t|y_{1:t}) = \int p(\boldsymbol{\beta}_t|\boldsymbol{\theta}_t, y_{1:t}) \times p(\boldsymbol{\theta}_t|y_{1:t}) \mathrm{d}\boldsymbol{\theta}_t. \tag{3.21}$$

It can be shown that $\boldsymbol{\beta}_t|y_{1:t} \sim \mathbb{C}\mathcal{N}\big(\boldsymbol{\beta}^+_t, \mathbf{G}^+_t\big)$, where $\boldsymbol{\beta}^+_t$ and $\mathbf{G}^+_t$ are given by the measurement update equations corresponding to the Kalman filter. In this manner, the filter jointly estimates the multiple target positions and velocities, using Monte Carlo based approach with one particle filter per target, and channel state, using a Kalman filter. We refer to this Bayesian filter as Multiple Rao-Blackwellized particle filter (MRBPF). The overall algorithm is given in Algorithm 2.

Algorithm 2 MRBPF Algorithm for Joint Tracking of Target and Channel States.

1: Initialize $\{\boldsymbol{\theta}_{k,0}^{(i)}\}_{i=1}^{N_\mathrm{s}}, \boldsymbol{\beta}_0^+$, and $\{\mathbf{G}_0^+\}$

2: Initialize the weights $w_{k,0}^{(i)} = 1/N_\mathrm{s}$ for $k = 1, 2, \cdots K$

3: **for** $t = 1 : T$ **do**

4: **parfor** $k = 1 : K$ **do**

5: Draw the samples $\{\boldsymbol{\theta}_{k,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ from the proposal distributions $p(\boldsymbol{\theta}_{k,t}|\boldsymbol{\theta}_{k,t-1})$.

6: Compute $\boldsymbol{\Phi}_t^{(i)}$ based on the samples $\{\boldsymbol{\theta}_{k,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}, k = 1, \cdots M$

7: *Kalman filter time update:* Compute the Kalman gain and obtain the predicted value of the channel state vector $\boldsymbol{\beta}_t^-$ using the following equations.

$$\begin{aligned}
\mathbf{G}_t^- &= \boldsymbol{\Sigma}_{\mathrm{v}_\beta} + \mathbf{G}_{t-1}^+ \\
\mathbf{P}_t^{(i)} &= \boldsymbol{\Sigma}_\mathrm{w} + \boldsymbol{\Phi}_t^{(i)} \mathbf{G}_t^- (\boldsymbol{\Phi}_t^{(i)})^H \\
\mathbf{J}_t^{(i)} &= \mathbf{G}_t^- (\boldsymbol{\Phi}_t^{(i)})^H (\mathbf{P}_t^{(i)})^{-1} \\
\boldsymbol{\beta}_t^- &= \boldsymbol{\beta}_{t-1}^+
\end{aligned}$$

8: *Particle filter time update:* Compute the predicted value of the k^{th} target state using the equation

$$\boldsymbol{\theta}_{k,t}^- = \sum_{i=1}^{N_\mathrm{s}} w_{k,t-1}^{(i)} \boldsymbol{\theta}_{k,t}^{(i)}$$

9: *Particle filter measurement update:* Compute the unnormalized weights for each k using the equation

$$\tilde{w}_{k,t}^{(i)} = w_{k,t-1}^{(i)} p\Big(\boldsymbol{y}_t | \boldsymbol{\theta}_{k,t}^{(i)}, \boldsymbol{\theta}_{-k,t}^-, \boldsymbol{\beta}_t^-\Big)$$

10: Normalize the weights using $w_{k,t}^{(i)} = \frac{\tilde{w}_{k,t}^{(i)}}{\sum_{i=1}^{N_\mathrm{s}} \tilde{w}_{k,t}^{(i)}}$

11: *Kalman filter measurement update:* Update the covariance matrix of the channel state and the channel state itself using

$$\begin{aligned}
\boldsymbol{z}_t^{(i)} &= \boldsymbol{\Phi}_k^{(i)} \boldsymbol{\beta}_k^- \\
\boldsymbol{\beta}_t^+ &= \frac{1}{M} \sum_{m=1}^{M} \sum_{i=1}^{N_\mathrm{s}} w_{m,k}^{(i)} \Big(\boldsymbol{\beta}_k^- + \mathbf{J}_t^{(i)} (\boldsymbol{y}_k - \boldsymbol{z}_k^{(i)}) \Big) \\
\mathbf{G}_t^+ &= \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{N_\mathrm{s}} w_{k,t}^{(i)} (\boldsymbol{I} - \mathbf{J}_t^{(i)} \boldsymbol{\Phi}_t^{(i)}) \mathbf{G}_t^-
\end{aligned}$$

12: **end for**

13: **end for**

3.4 Antenna Scheduling and Power Allocation

We use the posterior Cramér-Rao bound (PCRB) as an optimization criterion for antenna scheduling and power allocation. The PCRB is a lower bound on the mean square error (MSE) of the Bayesian estimates of the state vector and hence we seek to find the optimal antenna set and the corresponding power to be transmitted by these antennas by minimizing the PCRB. Another motivation for using the PCRB is that it can be computed in a sequential manner [66] in every interval. The recursive formulation for the computation of PCRB suits the problem of target tracking, where we need to find the estimates of the state vector in every tracking interval, and hence the PCRB is a natural choice for the optimization criterion. The PCRB for the tracking problem is defined and derived as follows.

3.4.1 Computation of the Posterior Cramér Rao Bound

Let $\Delta_{\boldsymbol{\eta}} = [\partial\eta_1, \partial\eta_2, \ldots \partial\eta_r]^T$ denote a vector of the partial derivatives with respect to the vector $\boldsymbol{\eta}$ and $\Delta_{\upsilon}^{\kappa} = \Delta_{\upsilon}\Delta_{\kappa}^{T}$ denote the partial derivative vectors. With this notation, the PCRB for an unbiased estimate of $\boldsymbol{\xi}$ has the form

$$\mathbb{E}[(\boldsymbol{\xi} - \hat{\boldsymbol{\xi}})(\boldsymbol{\xi} - \hat{\boldsymbol{\xi}})^T] \geq \mathbf{J}^{-1}, \tag{3.22}$$

where $\mathbf{J}$ is the Fisher information matrix (FIM), given as

$$\mathbf{J} = -\mathbb{E}\Big[\Delta_{\xi}^{\xi} \log\big(p(\boldsymbol{\xi}, \boldsymbol{y})\big)\Big]. \tag{3.23}$$

The recursive equation to compute the FIM in an online and recursive manner was proposed in [66], and we state it here for completeness.

Theorem 3. *The sequence* $\{\mathbf{J}_t\}$ *of posterior information sub-matrices for estimating state vector obeys the recursion*

$$\mathbf{J}_{t+1} = \boldsymbol{D}_t^{22} - \boldsymbol{D}_t^{21}(\mathbf{J}_t + \boldsymbol{D}_t^{11})^{-1}\boldsymbol{D}_t^{12}, \tag{3.24}$$

where

$$\boldsymbol{D}_t^{11} = \mathbb{E}\big\{ -\Delta_{\boldsymbol{\xi}_t}^{\boldsymbol{\xi}_t} \log p(\boldsymbol{\xi}_{t+1}|\boldsymbol{\xi}_t)\big\}$$

$$\boldsymbol{D}_t^{12} = \mathbb{E}\big\{ -\Delta_{\boldsymbol{\xi}_{t+1}}^{\boldsymbol{\xi}_t} \log p(\boldsymbol{\xi}_{t+1}|\boldsymbol{\xi}_t)\big\}$$

$$\boldsymbol{D}_t^{21} = \mathbb{E}\big\{ -\Delta_{\boldsymbol{\xi}_t}^{\boldsymbol{\xi}_{t+1}} \log p(\boldsymbol{\xi}_{t+1}|\boldsymbol{\xi}_t)\big\} \textit{and}$$

$$\boldsymbol{D}_t^{22} = \mathbb{E}\big\{ -\Delta_{\boldsymbol{\xi}_{t+1}}^{\boldsymbol{\xi}_{t+1}} \log p(\boldsymbol{\xi}_{t+1}|\boldsymbol{\xi}_t)\big\} + \mathbb{E}\big\{ -\Delta_{\boldsymbol{\xi}_{t+1}}^{\boldsymbol{\xi}_{t+1}} \log p(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{t+1})\big\},$$

and the expectation is taken with respect to the joint distribution $p(\boldsymbol{\xi}_{t+1}, \boldsymbol{y}_{t+1})$.

From Eq. (3.4), $p(\boldsymbol{\xi}_{t+1}|\boldsymbol{\xi}_t) \sim \mathcal{N}(\boldsymbol{F}\boldsymbol{\xi}_t, \boldsymbol{\Sigma}_{\mathrm{v}_\xi})$. With this substitution and using the matrix-inversion lemma, it can be shown that Eq. (3.24) reduces to

$$\mathbf{J}_{t+1} = \left[\boldsymbol{\Sigma}_{\mathrm{v}_\xi} + \boldsymbol{F}\mathbf{J}_t^{-1}\boldsymbol{F}^T\right]^{-1} + \boldsymbol{\Gamma}_{t+1}, \tag{3.25}$$

where

$$\boldsymbol{\Gamma}_{t+1} = \mathbb{E}\left[-\Delta_{\boldsymbol{\xi}_{t+1}}^{\boldsymbol{\xi}_{t+1}} \log p(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{t+1})\right]. \tag{3.26}$$

Since the estimate of the state vector, $\boldsymbol{\xi}_{t+1}$, is not available at time t, the term $\boldsymbol{\Gamma}_{t+1}$ does not have a closed form expression. We use Monte Carlo sampling to obtain an

approximate value as outlined below.

$$\begin{aligned}\boldsymbol{\Gamma}_{t+1} &= \mathbb{E}\left[-\Delta_{\xi_{t+1}}^{\xi_{t+1}} \log p(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{t+1})\right] \\ &= \mathbb{E}_{\xi}\left[\mathbb{E}_{\boldsymbol{y}|\xi}\left[-\Delta_{\xi_{t+1}}^{\xi_{t+1}} \log p(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{t+1})\right]\right] \\ &\approx \frac{1}{K}\sum_{i=1}^{\mathrm{N_s}}\sum_{k=1}^{K} w_{k,t}^{(i)} \boldsymbol{\Xi}_{t+1}\left(\boldsymbol{\theta}_{t+1}^{(i)}, \boldsymbol{\beta}_{t+1}^{(i)}\right), \end{aligned} \tag{3.27}$$

where $\boldsymbol{\Xi}_{t+1}\left(\boldsymbol{\theta}_{t+1}, \boldsymbol{\beta}_{t+1},\right) = \mathbb{E}_{\boldsymbol{y}|\xi}\left[-\Delta_{\xi_{t+1}}^{\xi_{t+1}} \log p(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{t+1})\right]$, $\boldsymbol{\xi}_{t+1} = \left[\boldsymbol{\theta}_{t+1}^T, \boldsymbol{\beta}_{t+1}^T\right]^T$, and $\boldsymbol{\theta}_{t+1}^{(i)}$ and $\boldsymbol{\beta}_{t+1}^{(i)}$ are the samples drawn from the state transition functions of the target state and channel state vectors following Eq. (3.1) and Eq. (3.2), respectively. The value of $\boldsymbol{\Xi}_{t+1}\left(\boldsymbol{\theta}_{t+1}, \boldsymbol{\beta}_{t+1},\right)$ is stated next.

Theorem 4. *Let $\pi_t \subseteq \{1, 2, \cdots N\}$ denote the set of the radars that are in operation at time t. For the measurement model described in Eq. (3.14), we have*

$$\boldsymbol{\Xi}_{t+1}\left(\boldsymbol{\theta}_{t+1}, \boldsymbol{\beta}_{t+1},\right) = 2\sum_{p,q,r\in\pi_{t+1}} \sqrt{\gamma_q\gamma_r}\Re\left\{\left(\frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \boldsymbol{\xi}_{t+1}}\right)\boldsymbol{\Sigma}_{\mathrm{w},p}^{-1}\left(\frac{\partial \boldsymbol{\mu}_{t+1}^{pr}}{\partial \boldsymbol{\xi}_{t+1}}\right)^H\right\}, \tag{3.28}$$

where $\boldsymbol{\mu}_{t+1}^{pq} = \sum_{k=1}^{K} \boldsymbol{\Phi}_{pqk,t+1}\boldsymbol{\beta}_{pqk,t+1}$.

The proof of the theorem is shown in Appendix A and he expressions for $\frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \boldsymbol{\xi}_{t+1}}$ are derived in Appendix B.

3.4.2 Approximate Greedy Algorithm for Adaptive Antenna Scheduling and Power Allocation

Our approach to antenna scheduling and power allocation for the radar network is based on the minimization of the predicted value of the PCRB, under suitable constraints. The constraints represent the bounds on the total power and total cost available for deploying the antennas. In general, the cost of communicating the information from a radar to the fusion center is proportional to the distance between them. Hence we use the Euclidean distance measure as an indicator of the communication cost. We devise the following constrained joint optimization problem for scheduling and power allocation.

$$\begin{aligned}
&\boldsymbol{\pi}^*_{t+1}, \boldsymbol{\gamma}^*_{t+1} = \arg\min_{\Pi,\Gamma} \operatorname{trace}(\mathbf{J}_{t+1})^{-1} \text{ subject to} \\
&\sum_{p\in\pi_{t+1}} \sqrt{(x_p^2 + y_p^2)} \leq \eta_{\mathrm{c}}, \text{ and} \\
&\sum_{p\in\pi_{t+1}} \gamma_p \leq \eta_{\mathrm{e}}
\end{aligned} \tag{3.29}$$

The first constraint in the problem represents the communication cost constraint and second constraint represents the power constraint. The parameters η_{c} and η_{e} correspond to the bounds on the total communication cost and the total power. Obtaining a solution to this joint optimization problem is NP-hard. We propose a two pass greedy algorithm to find a suboptimal solution to this problem. We separate the problem into two parts: the problem of finding the antennas to be employed and the problem of finding the power to be allocated to these antennas. In the first pass, we transmit equal power on all the antennas and solve the problem of selecting an optimal set of antennas to be used. To select an optimal subset of antennas to be

used, we can evaluate the FIM over all possible combinations of subsets of antennas, and choose the best one. The complexity of such an evaluation grows exponentially with the number of antennas. We obtain an approximate solution by employing an approximate greedy algorithm whose computational complexity grows linearly with the number of antennas. We compute the FIM for all the antennas separately, and greedily select the ones which minimize the product of the Euclidian distance and the trace of the inverse of the FIM. Once the antennas are selected, in the second pass, we distribute the power to these antennas, again using a greedy approach. In this case, we allocate more power to the antennas that maximize the overall signal to noise ratio (SNR). Since the PCRB is inversely related to the SNR [15], we are minimizing by PCRB by maximizing the SNR. The algorithm is summarized below in 3.

3.5 Numerical Results

In this section, we use numerical examples to study the performance of the proposed cognitive radar network system in the presence of the time-varying multipath propagation conditions. We demonstrate the advantage of the proposed MRBPF method by comparing it to the SPF. We also demonstrate the advantage of the proposed adaptive antenna scheduling and power allocation methods compared to the fixed antenna scheme and equal power allocation. Finally, we demonstrate the advantage of the multipath modeling. We describe the simulation setup first and then discuss the numerical examples.

Signal and Multipath parameters: We considered OFDM waveforms with eight ($G = 8$) sub-carriers loaded with same symbol in all the sub-carriers. The total bandwidth

Algorithm 3 Greedy Algorithm for Sensor Scheduling and Power Allocation.

for t = 1:T **do**
  First Pass:
  Initialize $\pi_{t+1} = \{1\}$
  for $i = 1 : P$ **do**
    Set $\pi_{t+1} = \{1, i\}$ and $\gamma_{t+1} = \frac{\eta_e}{P}$.
    Compute $e_i = \text{trace}(\mathbf{J}_{k+1}^{-1})$
    Compute the distance between the i^{th} antenna and the first antenna i.e., $d_i = \sqrt{(x_i^2 + y_i^2)}$
    Compute the product $v_i = d_i \times e_i$
  end for
  Sort the antennas in decreasing order of $\boldsymbol{v}$. Let the sorted list be stored in SortedList
  while $\sum_{i=1}^{j} d_i < \eta_c$ and SortedList is non-empty **do**
    Select antenna i sequentially from SortedList
    $\pi_{t+1} = \pi_{t+1} \cup \{i\}$
  end while
  If $e_{j+1} > e_{\pi_{t+1}}$, where $e_{\pi_{t+1}} = \text{trace}(\mathbf{J}_{t+1}^{-1})$ with π_{t+1}, choose $\pi_{t+1}^* = \pi_{t+1}$, otherwise, choose $\pi_{t+1}^* = \{1, j+1\}$
  Second Pass:
  for $i \in \pi_{k+1}^*$ **do**
    Compute the ratio

$$\lambda_i = \sum_{p \in \pi_{k+1}^*} \frac{\sum_{k=1}^{K} [\mathbf{\Sigma}_\beta]_{ip}}{\sigma_{\text{w},p}^2} \tag{3.30}$$

    Choose the energy to be transmitted on the i^{th} antenna as

$$\gamma_{i,k+1}^* = \frac{\lambda_i \eta_e}{\sum_{j \in \pi_{k+1}^*} \lambda_j} \tag{3.31}$$

  end for
  Update FIM
  Update $\mathbf{J}_{t+1}$ using Eqs. (3.25), (3.27), (3.28) and $\boldsymbol{\pi}_{t+1}^*$ and $\boldsymbol{\gamma}_{t+1}^*$
end for

was 100Mhz ($B = 100$) and the carrier frequency, f_c, of the transmitted waveforms was 1Ghz. We used four ($L = 4$) pulses in each tracking interval. The multipath environment consisted of delay and Doppler shifts. We used three Doppler shifts and two delay shifts, i.e., $N_\tau = 2$ and $N_f = 3$. The vector $\boldsymbol{\beta}_{pqk}$ was generated from a Gaussian distribution with zero mean and unit variance and scaled later such that variance of the coefficients corresponding to different delays decayed exponentially.

Target and the Radar Network parameters: We considered three different configurations for the target trajectories and the antenna locations for the examples. These configurations are shown in Fig. 3.1.

In the first configuration, the network consisted of three monostatic radars located at

$$(x_1, y_1) = (0, 0), (x_2, y_2) = (10, 25), (x_3, y_3) = (25, 20).$$

There were two crossing targets ($K = 2$) moving in the region of interest with the initial position of the first target at $(x_1, y_1) = (5, 5)$ m and that of the second target at $(x_2, y_2) = (30, 10)$ m. The targets were moving with constant velocities of $(\dot{x}_1, \dot{y}_1) = (12, 12)$ m/s and $(\dot{x}_2, \dot{y}_2) = (-15, 15)$ m/s along linear trajectories. The co-variance matrix of the process noise for the target state transition was given by Eq. (2.12) with with $\epsilon_1 = \epsilon_2 = 1$. The co-variance matrix of the process noise for the channel state transition was given by Eq. (3.3) with $\boldsymbol{\Sigma}_\beta = \sigma_\beta^2 \boldsymbol{I}_P$, where $\sigma_\beta^2 = 5 \times 10^{-2}$. The variance of the measurement noise at each receiver was $\sigma_{\mathrm{w},p}^2 = 5 \times 10^{-2}, p = 1, \cdots P$.

In the second configuration, the network consisted of eight monostatic radars located at

$$(x_1, y_1) = (0, 0), (x_2, y_2) = (3, 3), (x_3, y_3) = (6, 6), (x_4, y_4) = (9, 9),$$

$$(x_5, y_5) = (12, 12), (x_6, y_6) = (15, 15), (x_7, y_7) = (18, 18), (x_8, y_8) = (21, 21).$$

There were two non-crossing targets ($K = 2$) with the initial position of the first target at $(x_1, y_1) = (0, 3)$ m and the second target at $(x_2, y_2) = (6, 0)$ m. The targets moved with constant velocities of $(\dot{x}_1, \dot{y}_1) = (0, 12)$ m/s and $(\dot{x}_2, \dot{y}_2) = (15, 0)$ m/s along linear trajectories. The co-variance matrix of the process noise for the target state transition was same as the corresponding co-variance matrix used in the first configuration. The co-variance matrix of the process noise for the channel state was given by Eq. (3.3), with

$$\mathbf{\Sigma}_\beta = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.005 & 0.005 & 0.005 & 0.003 & 0.002 \\ 0.01 & 0.01 & 0.01 & 0.005 & 0.005 & 0.005 & 0.003 & 0.002 \\ 0.01 & 0.01 & 0.01 & 0.005 & 0.005 & 0.005 & 0.003 & 0.002 \\ 0.005 & 0.005 & 0.005 & 0.01 & 0.005 & 0.005 & 0.003 & 0.002 \\ 0.005 & 0.005 & 0.005 & 0.005 & 0.01 & 0.005 & 0.003 & 0.002 \\ 0.005 & 0.005 & 0.005 & 0.005 & 0.005 & 0.01 & 0.003 & 0.002 \\ 0.003 & 0.003 & 0.003 & 0.003 & 0.003 & 0.003 & 0.01 & 0.002 \\ 0.002 & 0.002 & 0.002 & 0.002 & 0.002 & 0.002 & 0.002 & 0.01 \end{bmatrix}$$

The variance of the measurement noise was given by the vector

$$\sigma_{\mathrm{w}}^2 = (\sigma_{\mathrm{w},1}^2, \cdots \sigma_{\mathrm{w},P}^2) = (5{\times}10^{-2}, 5{\times}10^{-2}, 3{\times}10^{-2}, 3{\times}10^{-2}, 5{\times}10^{-2}, 2{\times}10^{-2} 2{\times}10^{-2}, 1{\times}10^{-2}).$$

In the third configuration, the network consisted of nine radars located at

$$(x_1, y_1) = (0, 0), (x_2, y_2) = (0, 20), (x_3, y_3) = (0, 40), (x_4, y_4) = (15, 0),$$

$(x_5, y_5) = (15, 20), (x_6, y_6) = (15, 40), (x_7, y_7) = (30, 0), (x_8, y_8) = (30, 20), (x_9, y_9) = (30, 40).$

There were five crossing targets ($K = 5$) with the initial position of targets at $(x_1, y_1) = (6, 5)$ m, $(x_2, y_2) = (25, 0)$ m, $(x_3, y_3) = (28, 30)$ m, $(x_4, y_4) = (10, 35)$ m, $(x_5, y_5) = (5, 5)$ m, respectively. The targets moved with constant velocities of $(\dot{x}_1, \dot{y}_1) = (12, 0)$ m/s, $(\dot{x}_2, \dot{y}_2) = (0, 18)$ m/s, $(\dot{x}_3, \dot{y}_3) = (-12, 0)$ m/s, $(\dot{x}_4, \dot{y}_4) = (0, -18)$ m/s, and $(\dot{x}_5, \dot{y}_5) = (12, 12)$ m/s, respectively, along linear trajectories. The co-variance matrix of the process noise for the target state transition was same as the one used for the first two configurations and the co-variance matrix of the process noise for the channel state was given by Eq. (3.3), with $\mathbf{\Sigma}_\beta = \sigma_\beta^2 \boldsymbol{I}_P$, where $\sigma_\beta^2 = 5 \times 10^{-2}$. The variance of the measurement noise was given by the vector

$$(\sigma_{\mathrm{w},1}^2, \cdots \sigma_{\mathrm{w},P}^2) = (5 \times 10^{-2}, 3 \times 10^{-2}, 5 \times 10^{-2}, 3 \times 10^{-1},$$

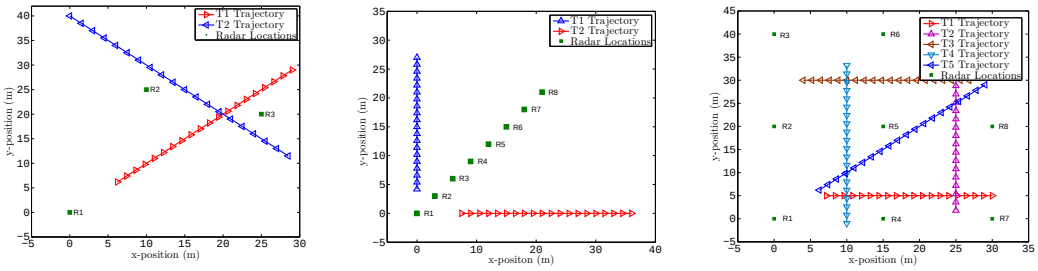
$$1 \times 10^{-2}, 3 \times 10^{-2}, 5 \times 10^{-2}, 3 \times 10^{-1}, 5 \times 10^{-1}).$$

In all the examples, the tracking interval length was chosen to be 0.1 seconds and the motion of the targets over 20 tracking intervals was considered. The parameter η_{e} was chosen to be 120, which corresponds to the total transmit power constraint. The simulations were averaged over 100 Monte Carlo iterations ($N_{\mathrm{mc}} = 100$). In order to analyze the performance improvement due to the adaptive scheduling and power allocation methods, we plot the composite root mean-squared error (CRMSE) vs the tracking interval index. We define the CMRSE in the range and velocity estimates, respectively, as

$$\text{CRMSE}_{\text{ran}} = \frac{1}{KN_{\text{mc}}} \sum_{i=1}^{N_{\text{mc}}} \sum_{k=1}^{K} \sqrt{(x_k - \hat{x}_{k,i})^2 + (y_k - \hat{y}_{k,i})^2},$$
$$\text{CRMSE}_{\text{vel}} = \frac{1}{KN_{\text{mc}}} \sum_{i=1}^{N_{\text{mc}}} \sum_{k=1}^{K} \sqrt{(\dot{x}_k - \hat{\dot{x}}_{k,i})^2 + (\dot{y}_k - \hat{\dot{y}}_{k,i})^2}, \tag{3.32}$$

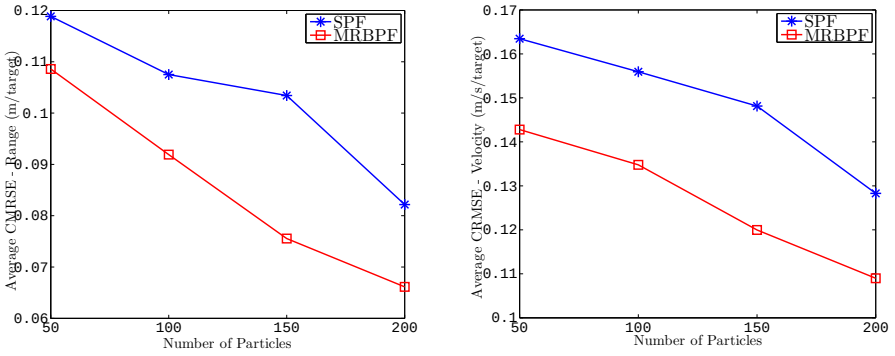
where $[\hat{x}_{k,i}, \hat{y}_{k,i}, \hat{\dot{x}}_{k,i}, \hat{\dot{y}}_{k,i}]^T$ is the estimate of the k^{th} target state in the i^{th} Monte-carlo run, and $[x_k, y_k, \dot{x}_k, \dot{y}_k]^T$ is the actual k^{th} target state.

Example 1: In this example, we demonstrate the advantage of the proposed multiple Rao-Blackwellized particle filtering method. We considered the first configuration for this example. In Fig. 3.2, we plot the composite root mean-squared error (CRMSE) averaged over all the 20 tracking intervals as a function of number of the particles for both SPF and the proposed MRBPF. It can be seen from the figure that MRBPF-based filtering resulted in lower CRMSE compared to the SPF-based filtering. This performance improvement is obtained since the MRBPF partitions the state space and computes the actual estimates of the channel state, and it updates the weight of individual target states instead of the joint target state. It can also be seen that we can achieve a given performance level using a MRBPF with fewer particles instead of a SPF with more particles. For example, to obtain an average CMRSE of 0.1 m/target we need approximately 80 particles using an MRBPF whereas we need around 160 particles using a SPF. Hence, MRBPF is computationally less expensive compared to the SPF, since we can get similar performance to that of SPF using fewer particles.



(a) First Configuration (b) Second Configuration (c) Third Configuration

Figure 3.1: Three configurations used in the numerical examples.



(a) CRMSE in Range (b) CRMSE in Velocity

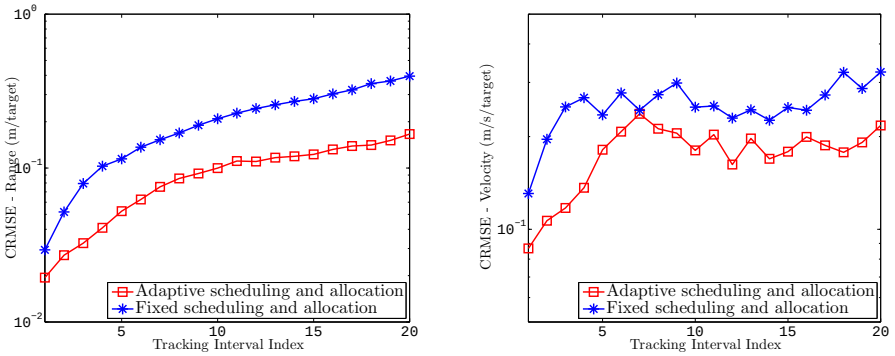
Figure 3.2: Average CRMSE of the range and the velocity estimates plotted against the number of particles for the SPF and the MRBPF.

Table 3.1: Table showing output of the antenna scheduling for one Monte Carlo Iteration.

Tracking Interval Index	# of Antennas (second configuration)	Antennas selected (second configuration)	# of Antennas (third configuration)	Antennas selected (third configuration)
1	4	1,2,3,4	5	1,6,2,3,4
2	4	1,2,3,4	5	1,8,6,2,4
3	4	1,2,3,4	5	1,8,2,3,4
4	3	1,4,7	5	1,8,2,3,4
5	3	1,7,4	5	1,8,2,3,4
6	3	1,7,4	5	1,8,3,2,4
7	3	1,7,4	5	1,8,3,2,4
8	3	1,7,4	5	1,8,3,2,4
9	3	1,7,4	4	1,2,5,6
10	3	1,7,4	4	1,8,3,7
11	3	1,7,4	4	1,8,3,7
12	3	1,7,4	4	1,7,3,6
13	3	1,7,4	4	1,6,3,7
14	3	1,7,4	4	1,6,3,7
15	3	1,8,3	5	1,8,6,2,4
16	3	1,8,3	5	1,8,6,2,4
17	2	1,7	2	1,6
18	2	1,7	2	1,6
19	2	1,8	2	1,6
20	2	1,8	5	1,6,7,2,4

Example 2: In this example, we demonstrate the advantage of the adaptive scheduling and resource allocation methods. We used the second configuration with $N_s = 200$ particles for this example. The parameter η_c was chosen to be 40. In Fig. 3.3, we plot the CMRSE in the range and the velocity estimates for this configuration. For the fixed scheduling and resource allocation, we used antennas $\{1, 2, 4\}$ and distributed the available power equally among them. We used three antennas so that the average number of antennas that are used remain same for both the adaptive case and the non-adaptive case. Using adaptive scheduling, four antennas were selected initially (see Table 3.1). Since the RMSE is inversely proportional to the number of the antennas, maximum number of antennas were used within the distance constraint. As the target moved away from the fusion center, the antennas that are closer to the target are used, although this increased the communication cost. As a result only two antennas were selected after a few iterations. As it can be seen, the performance using adaptive scheduling and resource allocation was better compared to the performance obtained using the fixed scheduling resource allocation.

Example 3: In this example, we used third configuration with $N_s = 200$ particles and compared the performance of cognitive radar employing adaptive scheduling and resource allocation with the performance of the standard radar that employed fixed scheduling. The parameter η_c was chosen to be 120 for this example. In Fig. 3.4, we plot the CRMSE in the range and the velocity estimates of both the targets for this configuration. For the fixed scheduling, we used antenna set $\{1, 3, 7, 9\}$. As it can be seen the performance using adaptive scheduling and resource allocation was better compared to the performance obtained using the fixed scheduling resource allocation.



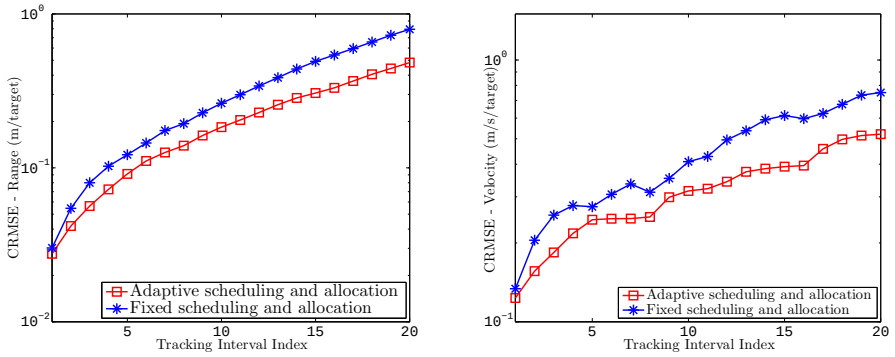
(a) CRMSE in Range (b) CRMSE in Velocity

Figure 3.3: Performance comparison with and without adaptive scheduling and power allocation for the second configuration.

Example 4: In this example, we demonstrate the advantage of the multipath modeling in the system. In Fig. 3.5, we plot the CRMSE in the range and the velocity estimates obtained using the MRBPF tracking with and without considering the multipath modeling. We used first configuration for this example with $N_s = 200$ particles. It can be seen that when the multipath model was not considered the CRMSE increased. For the same parameters, the performance by considering the effect of time-varying multipath channel model was significantly better. This is due to the additional degrees of freedom that an urban environment provides in the form of delay and Doppler diversity. When the receiver has information about the propagation conditions, it can exploit the multipath nature of the urban environment to obtain a better performance.

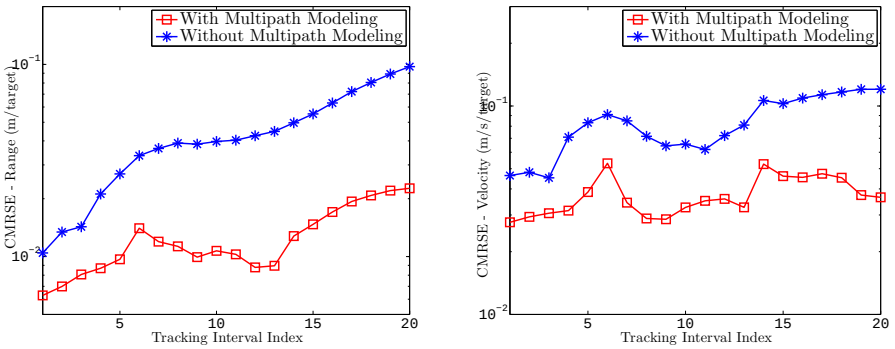
3.6 Summary

We considered the problem of multiple target tracking in a time-varying multipath channel which is characterized by an unknown channel state. We proposed a new



(a) CRMSE in Range (b) CRMSE in Velocity

Figure 3.4: Performance comparison with and without adaptive scheduling and power allocation for the third configuration.



(a) CRMSE in Range (b) CRMSE in Velocity

Figure 3.5: Performance comparison with and without multipath modeling.

hybrid filter called the Multiple Rao-Blackwell particle filter (MRBPF) for the joint estimation of the target state and the channel state. The proposed filter was efficient for tracking high-dimensional state vector as it operates by partitioning the state space into lower dimensional subspaces. We then computed the posterior Cramér-Rao bound (PCRB) on the estimates of the target and the channel states and used it as an optimization criterion to find the sensors to be employed at each time and the power to be transmitted on them. Since the optimal solution to the sensor scheduling and power allocation problem is NP-hard, we proposed a sub-optimal, but computationally efficient, method for scheduling and power allocation based on greedy programming.

Chapter 4

Hierarchical Particle Filtering using Multi-Modal Sensors[7]

In this chapter, we develop a filtering technique for tracking an unknown number and unknown types of targets moving in the region of interest.

4.1 Introduction

The last few decades have seen burgeoning growth in the development of inexpensive sensing devices due to the advances in the micro-electro-mechanical system (MEMS) fabrication techniques. With sensing devices becoming cheaper, it is now possible to deploy several kinds of sensors to observe the unknown state through the measurements collected by the sensors. Such networks are called multi-modal sensor networks, and they are deployed for various applications including visual tracking [67], cardiovascular diagnosis [68], 3D image reconstruction [69], temperature inference [70], and

[7]Based on P. Chavali and A. Nehorai, "Hierarchical Particle Filtering for Multi-Modal Sensor Networks With Application to Multiple-Target Tracking", to appear in *Signal Processing*.

target tracking [1]. In multi-modal sensor networks, different quantities associated with the same state are measured using sensors of different kinds. A single sensor is usually unable to provide complete information about the hidden state, and therefore it is necessary to combine the information provided by multiple sensors for obtaining an estimate of the state. When complimentary information from different kinds of sensors is appropriately combined, the performance of the overall system improves significantly compared with the performance of each modality separately.

The problem of combining the diverse and sometimes inconsistent measurements provided by multiple sensors is called data fusion [71], [72], [73]. Fusing the data acquired from different sensors has to be done before the estimation process and therefore, it is a critical bottleneck. There are several data fusion algorithms that are proposed in the literature which combine the data at one of the following three levels. At the low level, the raw data obtained from each sensor is combined. This kind of data fusion is used when the employed sensors collect similar types of data. At the intermediate level, the raw data from each sensor is processed to obtain the features of interest, and the features are combined. The problem with this approach is that the process of feature selection is subjective and problem dependent. At the high level or the decision level, the posterior distributions of the unknown state vector, which are computed using the raw data from individual sensors, are combined. Graphical models [74] have been used as suitable candidates for data fusion at the high level.

In this chapter, we first derive the weight update equations for a particle filter that employs (i) linear opinion pool (ii) independent opinion pool and (iii) independent likelihood pool (described in the next section) for combining the information from the various sensing modalities. In all these three techniques, the information from the

multiple sensing modalities is combined only in the measurement-update equation 1.4.

Second, we propose a new particle filtering technique for a special class of models called the hierarchical models. We call this variation of the particle filter, a *hierarchical particle filter* (HPF). Hierarchical models are encountered frequently in practice when the observations about the unknown state are obtained using multi-modal sensors, where each sensor is used to obtain information about a specific aspect of the state vector. We will provide more examples of such models in the subsequent section. Our proposed filtering method provides two advantages when compared to other methods that are described in the previous subsection. We will discuss these advantages in the subsequent sections.

We then formulate the problem of tracking an unknown number of targets using multi-modal measurements. We use the proposed HPF for the joint initiation, termination and tracking of multiple targets. We employ a multi-modal sensor network comprising a multistatic radar, an infrared camera, and a human scout, to obtain the information about the target scene. The unknown state vector comprises the number of targets, their positions, velocities, and categories. We describe the detailed state model for this system, and derive the measurement models for the three sensors that are used.

4.2 Existing Techniques for Data Fusion

Fusion of information from different modalities has been a major research area yet there has not been a widely accepted standard technique [73]. Given the state $\boldsymbol{\theta}_t$, we assume that the data collected is conditionally independent across the modalities,

i.e., $p(\boldsymbol{y}_t|\boldsymbol{\theta}_t) = \prod_{n=1}^{N} p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_t)$. Our goal is to estimate the global posterior filtered density $p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{1:t})$. The most common approach is to combine the local posterior densities in the measurement update step to compute the global posterior. In [75], the author proposes to compute the global posterior $p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{1:t})$, given the observations from each modality, as a linear combination of the local posterior distribution at each modality, i.e.,

$$p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{1:t}) = \sum_{n=1}^{N} \pi_n p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{n,1:t}), \tag{4.1}$$

where each weight π_n represents a subjective measure of the reliability of the information from the n^{th} modality and $\sum_{n=1}^{N} \pi_n = 1$. This method is called the linear opinion pool. One of the shortcomings of the linear opinion pool is its inability to reinforce opinion because the weights are usually unknown except in very specific applications. The independent opinion pool is a product form variation of the linear opinion pool and the global posterior is computed as the product of the local posteriors [76].

$$p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{1:t}) = \prod_{n=1}^{N} p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{n,1:t}). \tag{4.2}$$

The problem with the independent opinion pool is that when the prior information at each modality is common, the global posterior will reinforce the opinion through the product of the priors. Hence, this method is suitable only if the prior information at each modality is obtained independently. When all the modalities have common prior information, the global posterior can be obtained through an independent likelihood pool [76], which is obtained as

$$p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{1:t}) = p(\boldsymbol{\theta}_t) \prod_{n=1}^{N} p(\boldsymbol{y}_{n,1:t} \mid \boldsymbol{\theta}_t). \tag{4.3}$$

As it can be seen, an independent likelihood pool simply multiplies the likelihoods obtained from the individual sensors to compute the global posterior density. Consequently, the multi-modal characteristics of different sensors can be lost if there is a large difference among the numerical values of these likelihoods. In such scenarios, the performance improvement obtained through sensor fusion is not significant.

In [77], the authors propose a mixture kernel-based Bayesian filtering for multi-modal data. In this approach, the local posterior distribution corresponding to each modality is modeled as a mixture of Gaussian distributions. Using parametric and non-parametric mixture models for approximating the posterior distribution has been first proposed in [78], and the authors in [77] use this modeling to represent the local posterior distributions. The global posterior distribution is then represented as a mixture of the local posterior distributions, and this representation is preserved through iterations of the Bayesian filtering. At each iteration, the prior and the partial observation from each modality are combined using a 2-stage method scheme to construct the proposal distribution. The details of this method can be found in [67], [77],and [79]. The 2-stage sampling improves the sampling of the particles, and reduces the degeneracy problem. The mixture weights corresponding to the local posterior distributions are evaluated in proportion to the measurement confidence at each modality. However, representing the local posterior distributions as a mixture of the Gaussian distributions is possible only if all the local distributions are continuous probability densities. For a general case where a local posterior distribution is a discrete distribution, such a representation is not valid. Further, estimating the parameters of the Gaussian distributions and the mixture weights for each local posterior requires training data from each modality, which is not, in general, available.

4.3 Hierarchical Bayesian Models

We consider a class of systems in which the state equation and the measurement equation follow a hierarchical structure. The evolution model and the measurement model for such a system can be described by the following equations:

$$\begin{aligned}
\boldsymbol{\theta}_{1,t} &= g_1(\boldsymbol{\theta}_{1,t-1}, \boldsymbol{v}_{1,t-1}); & \boldsymbol{y}_{1,t} &= h_1(\boldsymbol{\theta}_{1,t}, \boldsymbol{w}_{1,t}), \\
\boldsymbol{\theta}_{2,t} &= g_2(\boldsymbol{\theta}_{2,t-1}, \boldsymbol{\theta}_{1,t}, \boldsymbol{v}_{2,t-1}); & \boldsymbol{y}_{2,t} &= h_2(\boldsymbol{\theta}_{1,t}, \boldsymbol{\theta}_{2,t}, \boldsymbol{w}_{2,t}), \\
&\vdots & &\vdots \\
\boldsymbol{\theta}_{N,t} &= g_N(\boldsymbol{\theta}_{N,t-1}, \boldsymbol{\theta}_{1:N-1,t}, \boldsymbol{v}_{N,t-1}); & \boldsymbol{y}_{N,t} &= h_N(\boldsymbol{\theta}_{1:N,t}, \boldsymbol{w}_{N,t}).
\end{aligned} \tag{4.4}$$

In the above, $\boldsymbol{\theta}_{1,t}, \ldots, \boldsymbol{\theta}_{N,t}$ are the N partitions of the state vector, $\boldsymbol{\theta}_{1:n,t} = \{\boldsymbol{\theta}_{1,t}, \ldots, \boldsymbol{\theta}_{n,t}\}$, and $\boldsymbol{y}_{1,t}, \ldots, \boldsymbol{y}_{N,t}$ are the partitions of the measurements. Each partition corresponds to the measurements obtained from one modality. Thus, it can be seen that the first modality observes only one partition of the state vector; the second modality observes two partitions; and finally, the last modality observes the entire state vector. Further, the evolution of each partition of the state vector depends on the previous partitions.

Multi-modal systems are typically characterized by such hierarchical observation and state models, because each sensing modality is selected to measure a specific aspect of the unknown state vector. For example, in surveillance systems that use a combination of audio and video sensors, the audio sensors determine the direction of arrival of the source, whereas the video sensors measure the number of sources in the region, and their positions. The measurements obtained by the audio sensors are, however, a function of the number of the sources and their positions. In the target tracking example that we considered, the human scout is trained to estimate the number of targets and to recognize their categories, but he is incapable of determining their

positions and velocities; the infrared camera sensors can obtain an estimate of the target locations, but they cannot determine their velocities; a multistatic radar can measure both the positions and velocities of the targets accurately, but it has limited capabilities in measuring the number of targets.

In some scenarios, each sensor might not be able to sense all the partitions of the state (see Example 2). Hierarchical modeling shown in Eq. 4.4 above will not describe such models. The proposed method can still be applied to such models, however the performance improvement obtained will not be significant.

4.4 Particle Filtering with Existing Fusion Methods

In this section, we derive the weight update equations for a standard particle filter that employs the linear opinion, the independent opinion, and the independent likelihood methods described in the earlier section.

4.4.1 Linear Opinion Pool

The posterior distribution $p(\boldsymbol{\theta}_{0:t} \mid \boldsymbol{y}_{1:t})$ is expressed as

$$p(\boldsymbol{\theta}_{0:t} \mid \boldsymbol{y}_{1:t}) \;=\; \sum_{n=1}^{N} \pi_n \sum_{i=1}^{N_{\mathrm{s}}} w_{n,t}^{(i)} \delta(\boldsymbol{\theta}_{0:t} - \boldsymbol{\theta}_{0:t}^{(i)}), \tag{4.5}$$

where $w_{n,t}^{(i)}$ is the weight corresponding to the i^{th} particle, and n^{th} modality. The weights corresponding to each modality are updated using the principle of importance

sampling.

$$\begin{aligned}
\tilde{w}_{n,t}^{(i)} &\propto \frac{p(\boldsymbol{\theta}_{0:t}^{(i)} \mid \boldsymbol{y}_{n,1:t})}{q(\boldsymbol{\theta}_{0:t}^{(i)} \mid \boldsymbol{y}_{n,1:t})} \\
&\propto \frac{p(\boldsymbol{\theta}_{0:t-1}^{(i)} \mid \boldsymbol{y}_{n,1:t-1})}{q(\boldsymbol{\theta}_{0:t-1}^{(i)} \mid \boldsymbol{y}_{n,1:t-1})} \times p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_t^{(i)}) \times \frac{p(\boldsymbol{\theta}_t^{(i)} \mid \boldsymbol{\theta}_{t-1}^{(i)})}{q(\boldsymbol{\theta}_t^{(i)} \mid \boldsymbol{\theta}_{0:t-1}^{(i)}, \boldsymbol{y}_{n,1:t})} \\
&\propto w_{n,t-1}^{(i)} \times p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_t^{(i)}) \times \frac{p(\boldsymbol{\theta}_t^{(i)} \mid \boldsymbol{\theta}_{t-1}^{(i)})}{q(\boldsymbol{\theta}_t^{(i)} \mid \boldsymbol{\theta}_{0:t-1}^{(i)}, \boldsymbol{y}_{n,1:t})} \\
&\propto w_{n,t-1}^{(i)} \times p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_t^{(i)}). \qquad (4.6)
\end{aligned}$$

The proposal distribution $q(\boldsymbol{\theta}_{0:t} \mid \boldsymbol{y}_{n,1:t})$ is chosen to factorize such that

$$q(\boldsymbol{\theta}_{0:t} \mid \boldsymbol{y}_{n,1:t}) = q(\boldsymbol{\theta}_{0:t-1} \mid \boldsymbol{y}_{n,1:t-1}) \times q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{0:t-1}, \boldsymbol{y}_{n,1:t}), \qquad (4.7)$$

and $q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}, \boldsymbol{y}_{n,t}) = p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1})$. Note that the proposal distribution is independent of the the sensor index n. Thus using the linear opinion pool, we update the weights due to each modality first, and then obtain the overall weights as

$$w_t^{(i)} = \sum_{n=1}^{N} \pi_n w_{n,t}^{(i)}. \qquad (4.8)$$

We refer to this filtering method as standard particle filtering with linear opinion pool (SPF-LO).

4.4.2 Independent Opinion Pool

Using Eq. (4.2), the global posterior distribution for the independent opinion pool is written as

$$p(\boldsymbol{\theta}_{0:t} \mid \boldsymbol{y}_{1:t}) \;\propto\; \sum_{i=1}^{N_{\mathrm{s}}} \prod_{n=1}^{N} \tilde{w}_{n,t}^{(i)} \delta(\boldsymbol{\theta}_{0:t} - \boldsymbol{\theta}_{0:t}^{(i)}). \tag{4.9}$$

From Eq. (4.9), it can be seen that the overall weights for standard particle filtering with independent opinion pool (SPF-IO) are given by

$$\tilde{w}_t^{(i)} \;\propto\; \prod_{n=1}^{N} \tilde{w}_{n,t}^{(i)}, \tag{4.10}$$

where $\{\tilde{w}_{n,t}^{(i)}\}_{i=1}^{N_{\mathrm{s}}}$ are the un-normalized weights obtained by following Eq. (4.6).

4.4.3 Independent Likelihood Pool

Using Eq. (4.3), the global posterior distribution for the independent likelihood pool is written as

$$p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{1:t}) \;=\; p(\boldsymbol{\theta}_t | y_{1:t-1}) \prod_{n=1}^{N} p(\boldsymbol{y}_{n,1:t} \mid \boldsymbol{\theta}_t). \tag{4.11}$$

Substituting Eq. (4.11) into the weight update equation defined by Eq. (1.18), it can be shown that the overall weights for standard particle filtering independent likelihood pool (SPF-IL) are given as

$$\tilde{w}_t^{(i)} \;\propto\; w_{t-1}^{(i)} \prod_{n=1}^{N} p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_t^{(i)}). \tag{4.12}$$

It should be noted here that for all the above methods, particle filters corresponding to all the sensors have the same set of particles.

4.5 Hierarchial Particle Filter

In this section, we describe the proposed hierarchical particle filter. The idea behind the hierarchical particle filter is to exploit the model structure in order to improve the efficiency of the sampling, thereby reducing the variance in the particle weights. We first partition the state space and the measurement space into lower-dimensional subspaces, and approximate the local posterior distribution corresponding to the first partition, given the measurements from the first partition, using a random particle set and the associated weights. To approximate the subsequent partitions, we generate the particles using the information obtained from the estimation of previous partitions, and update the weights using the measurements from the current partition. Thus, unlike the earlier approaches, we combine the information from different modalities in both the time-update step and the measurement-update step. We also describe the process of obtaining the particles from the proposal distribution, and updating their corresponding weights.

Let $\boldsymbol{\theta}_{1,t}, \ldots, \boldsymbol{\theta}_{N,t}$ be the N partitions of the state vector. We find the posterior distribution of each partition individually, by using the information provided from the estimates of the previous partitions and the measurements associated with the current partition. Without the loss generality, we assume that $\boldsymbol{\theta}_{1,t}$ is estimated first, followed by $\boldsymbol{\theta}_{2,t}$ and so on. The measurement partitions are labeled as $\boldsymbol{y}_{1,t}, \boldsymbol{y}_{2,t}, \ldots, \boldsymbol{y}_{N,t}$. In the first stage of the algorithm, we estimate the first partition of the state vector using the measurements from the first partition. We employ a standard particle filter

for approximating the local posterior distribution $p(\boldsymbol{\theta}_{1,0:t} \mid \boldsymbol{y}_{1,1:t})$. The particles are drawn from the proposal distribution $q_1(\boldsymbol{\theta}_{1,0:t} \mid \boldsymbol{y}_{1,1:t})$, which is chosen to be the state transition distribution corresponding to the first partition, and the weights are updated following Eq. (1.18). We denote by $\boldsymbol{\lambda}_{1,t}$, the discrete probability distribution defined by the normalized weights $\{w_{1,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ associated with the particles $\{\boldsymbol{\theta}_{1,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$.

In the second stage of the filtering method, we use the posterior probability distribution $\boldsymbol{\lambda}_{1,t}$ as extrinsic information for estimating the partition $\boldsymbol{\theta}_{2,t}$ along with the measurements. In other words, we are interested in evaluating the posterior distribution conditional on $\boldsymbol{\lambda}_{1,t}$, $p(\boldsymbol{\theta}_{2,0:t} \mid \boldsymbol{y}_{2,1:t}, \boldsymbol{\lambda}_{1,t})$. We sample the particles $\{\boldsymbol{\theta}_{2,t}\}_{i=1}^{N_\mathrm{s}}$ from the proposal distribution $q_2(\boldsymbol{\theta}_{2,0:t} \mid \boldsymbol{y}_{2,1:t}, \boldsymbol{\lambda}_{1,1:t})$, which depends also on $\boldsymbol{\lambda}_{1,t}$, and then evaluate the associated weights. In a similar way, for the n^{th} stage, the posterior probability distributions $\boldsymbol{\lambda}_{1:n-1,t} = \{\boldsymbol{\lambda}_{1,t}, \ldots, \boldsymbol{\lambda}_{n-1,t}\}$ are used as extrinsic information to draw the samples and update the corresponding weights of the n^{th} partition of the state vector. The posterior distribution for the n^{th} partition, $p(\boldsymbol{\theta}_{n,0:t} \mid \boldsymbol{y}_{n,1:t}, \boldsymbol{\lambda}_{1:n-1,1:t})$, is expressed as

$$
\begin{aligned}
& p(\boldsymbol{\theta}_{n,0:t} \mid \boldsymbol{y}_{n,1:t}, \boldsymbol{\lambda}_{1:n-1,1:t}) \\
= \; & p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,0:t}, \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t}) \times \frac{p(\boldsymbol{\theta}_{n,0:t} \mid \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t})}{p(\boldsymbol{y}_{n,t} \mid \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t})} \\
= \; & p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}, \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t}) \times \frac{p(\boldsymbol{\theta}_{n,t} \mid \boldsymbol{\theta}_{n,0:t-1}, \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t})}{p(\boldsymbol{y}_{n,t} \mid \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t})} \\
\times \; & p(\boldsymbol{\theta}_{n,0:t-1} \mid \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t}) \\
= \; & p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}, \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t}) \times \frac{p(\boldsymbol{\theta}_{n,t} \mid \boldsymbol{\theta}_{n,t-1}, \boldsymbol{\lambda}_{1:n-1,t})}{p(\boldsymbol{y}_{n,t} \mid \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t})} \\
\times \; & p(\boldsymbol{\theta}_{n,0:t-1} \mid \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t-1}) \\
\propto \; & p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}, \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t}) \times p(\boldsymbol{\theta}_{n,t} \mid \boldsymbol{\theta}_{n,t-1}, \boldsymbol{\lambda}_{1:n-1,t}) \\
\times \; & p(\boldsymbol{\theta}_{n,0:t-1} \mid \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t-1})
\end{aligned}
\tag{4.13}
$$

The proposal distribution for the n^{th} stage $q_n(\boldsymbol{\theta}_{n,0:t} \mid \boldsymbol{y}_{n,1:t}, \boldsymbol{\lambda}_{1:n-1,1:t})$ is chosen to factorize such that

$$q_n(\boldsymbol{\theta}_{n,0:t} \mid \boldsymbol{y}_{n,1:t}, \boldsymbol{\lambda}_{1:n-1,1:t}) = q_n(\boldsymbol{\theta}_{n,t} \mid \boldsymbol{\theta}_{n,t-1}, \boldsymbol{\lambda}_{1:n-1,t}) \times q_n(\boldsymbol{\theta}_{n,0:t-1} \mid \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t-1}). \tag{4.14}$$

Hence, the weight update equation for the n^{th} stage can be expressed as

$$\begin{aligned}
\tilde{w}_{n,t}^{(i)} &\propto \frac{p(\boldsymbol{\theta}_{n,0:t}^{(i)} \mid \boldsymbol{y}_{n,1:t}, \boldsymbol{\lambda}_{1:n-1,1:t})}{q_n(\boldsymbol{\theta}_{n,0:t}^{(i)} \mid \boldsymbol{y}_{n,1:t}, \boldsymbol{\lambda}_{1:n-1,1:t})}, \\
&\propto \frac{p(\boldsymbol{\theta}_{n,0:t-1}^{(i)} \mid \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t-1})}{q_n(\boldsymbol{\theta}_{n,0:t-1}^{(i)} \mid \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t-1})} \times \frac{p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}^{(i)}, \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t})}{q_n(\boldsymbol{\theta}_{n,t}^{(i)} \mid \boldsymbol{\theta}_{n,t-1}^{(i)}, \boldsymbol{\lambda}_{1:n-1,t})} \\
&\quad \times p(\boldsymbol{\theta}_{n,t}^{(i)} \mid \boldsymbol{\theta}_{n,t-1}^{(i)}, \boldsymbol{\lambda}_{1:n-1,t}), \\
&\propto w_{n,t-1}^{(i)} \times \frac{p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}^{(i)}, \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t})}{q_n(\boldsymbol{\theta}_{n,t}^{(i)} \mid \boldsymbol{\theta}_{n,t-1}^{(i)}, \boldsymbol{\lambda}_{1:n-1,t})} \times p(\boldsymbol{\theta}_{n,t}^{(i)} \mid \boldsymbol{\theta}_{n,t-1}^{(i)}, \boldsymbol{\lambda}_{1:n-1,t}),
\end{aligned} \tag{4.15}$$

where $\{\boldsymbol{\theta}_{n,t}^{(i)}\}_{i=1}^{N_s}$ are the samples drawn from the distribution.

4.5.1 Sampling from the Proposal Distribution

The choice of the proposal distribution is crucial and specific to the proposed hierarchical model. For optimal efficiency, it is necessary that the distribution proposes the most probable particles. In this work, we choose the proposal distribution for the n^{th} stage, $n > 1$, to be

$$q_n(\boldsymbol{\theta}_{n,t} \mid \boldsymbol{\theta}_{n,t-1}, \boldsymbol{\lambda}_{1:n-1,t}) = p(\boldsymbol{\theta}_{n,t} \mid \boldsymbol{\theta}_{n,t-1}, \boldsymbol{\lambda}_{1:n-1,t}). \tag{4.16}$$

This choice results in a weight update equation which is simple to implement. It is given as

$$\tilde{w}_{n,t}^{(i)} \propto w_{n,t-1}^{(i)} \times p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}^{(i)}, \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t}). \tag{4.17}$$

We make a comment here that sampling the particles from the above proposal distribution is different than sampling them from the distribution $p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1})$, even for hierarchical models that are described in Sec. 4.3. In our approach, using the proposal distribution described by Eq. (4.16), we obtain particles in stages. We use information provided by the weights corresponding to the samples from the first stage to obtain the samples for the subsequent stages. In this manner, we are combining the measurements obtained from earlier stages in both the time-update and the measurement-update steps. This is in contrast to the traditional sampling schemes which draw samples of all the partitions in one stage and do not use information provided by the measurements to obtain the particles.

In order to draw the samples from the distribution $p(\boldsymbol{\theta}_{n,t} \mid \boldsymbol{\theta}_{n,t-1}, \boldsymbol{\lambda}_{1:n-1,t})$, we use the following steps:

1. Choose N_s samples of $\boldsymbol{\theta}_{1,t}, \ldots, \boldsymbol{\theta}_{n-1,t}$, generated from the earlier $N-1$ stages.
2. Choose, for each partition, the samples $\{\boldsymbol{\theta}_{m,t}\}_{i=1}^{N_\mathrm{s}}, m = 2, \ldots, n-1$, with replacement, according to the probability distribution $\boldsymbol{\lambda}_{m,t}$.
3. Generate the samples $\{\boldsymbol{\theta}_{n,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ according to the known state transition probability $p(\boldsymbol{\theta}_{n,t} \mid \boldsymbol{\theta}_{1:n-1,t}, \boldsymbol{\theta}_{n,t-1})$.

As a result of the second step, the samples with higher weights are selected more frequently. Due to this selection, there is a lower probability of choosing a sample of other partitions that will lead to an incorrect sample of the current partition.

Therefore, the weights associated with the samples generated using this approach will have a lower variance.

4.5.2 Evaluating the Likelihood

In this subsection, we will describe how the likelihood function $p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}, \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,t})$ can be evaluated. We will start with an observation that the measurements $\boldsymbol{y}_{n,1:t-1}$ contain information about the partitions $\boldsymbol{\theta}_{1:n-1,t}$. Hence the likelihood

$$p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}, \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t}) \neq p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}, \boldsymbol{\lambda}_{1:n-1,1:t}). \tag{4.18}$$

In order to evaluate the likelihood, we use an approximation by replacing the probability distributions $\boldsymbol{\lambda}_{1:n-1,1:t}$ in $p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}, \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t})$ with samples $\{\boldsymbol{\theta}^{(i)}_{1:n-1,1:t}\}$ that are drawn according to these distributions. As a result, the likelihood simplifies as

$$\begin{aligned} p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}, \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\lambda}_{1:n-1,1:t}) &\approx p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}, \boldsymbol{y}_{n,1:t-1}, \boldsymbol{\theta}^{(i)}_{1:n-1,1:t}), \\ &= p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}, \boldsymbol{\theta}^{(i)}_{1:n-1,t}). \end{aligned} \tag{4.19}$$

The last equality in the above equation holds true since, given the state vector $\boldsymbol{\theta}_{1:n,t}$, the measurement $\boldsymbol{y}_{n,t}$ is independent of the other measurements $\boldsymbol{y}_{n,1:t-1}$. The samples $\boldsymbol{\theta}^{(i)}_{1:n-1,t}$ are chosen according to the distributions $\boldsymbol{\lambda}_{1:n-1,t}$, and the samples that are obtained during the sampling step can be reused here. The distribution $p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{n,t}, \boldsymbol{\theta}^{(i)}_{1:n-1,t})$ can then be evaluated using the observation model $p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{1:n,t})$.

4.5.3 Discussion

The proposed hierarchical approach has two advantages over the conventional particle filtering methods described in Section 4.4. First, by partitioning the state space into several subspaces, we reduce the effect of the degeneracy phenomenon. Second, since the proposal distribution for each stage is chosen using the information from the previous stages, sampling is more efficient and thereby, the particle weights will have a lower variance.

4.5.4 Examples

In this section, we present two examples to describe the performance of the proposed hierarchical filtering method. We consider a state vector which is partitioned into three subspaces. The measurements are collected by three modalities. Each partition of the state vector is of unit dimension, and the corresponding measurements are scalars.

First Example: In the first example, the scalar probability distributions governing the state equation and the measurement equations for each partition are given as

$$\begin{aligned}
p(y_{1,t} \mid \theta_{1,t}) &\sim \mathcal{N}(y_{1,t}; \theta_{1,t}^2, \sigma_{y_1}^2) \\
p(\theta_{1,t} \mid \theta_{1,t-1}) &\sim \mathcal{N}(\theta_{1,t}; 1 + \theta_{1,t-1}, \sigma_{\theta_1}^2) \\
p(y_{2,t} \mid \theta_{1,t}, \theta_{2,t}) &\sim \mathcal{N}(y_{2,t}; \sin(\theta_{1,t}) + \cos(\theta_{2,t}), \sigma_{y_2}^2) \\
p(\theta_{2,t} \mid \theta_{2,t-1}, \theta_{1,t}) &\sim \mathcal{N}(\theta_{2,t}; \theta_{2,t-1} + \theta_{1,t}, \sigma_{\theta_2}^2) \\
p(y_{3,t} \mid \theta_{1,t}, \theta_{2,t}, \theta_{3,t}) &\sim \mathcal{N}(y_{3,t}; \frac{\theta_{1,t}^2 + \theta_{2,t}^2 + \theta_{3,t}^2}{(\theta_{1,t} + \theta_{2,t} + \theta_{3,t})^2}, \sigma_{y_3}^2) \\
p(\theta_{3,t} \mid \theta_{3,t-1}, \theta_{2,t}, \theta_{1,t}) &\sim \mathcal{N}(\theta_{3,t}; \theta_{3,t-1} + \theta_{2,t} + \theta_{1,t}, \sigma_{\theta_3}^2), \qquad (4.20)
\end{aligned}$$

where $\sigma_{y_1}^2 = \sigma_{y_2}^2 = \sigma_{y_3}^2 = 0.2$, and $\sigma_{\theta_1}^2 = \sigma_{\theta_2}^2 = \sigma_{\theta_3}^2 = 0.1$. We used $N_\mathrm{s} = 200$ particles for the simulation and averaged the result for $N_\mathrm{c} = 50$ Monte-Carlo iterations. In Fig. 4.1, we plot the root mean-squared error (RMSE) for the three partitions. It can be seen that the HPF produces a lower RMSE when compared to the RMSE obtained using SPF-LO, SPF-IO and SPF-IL.

Second Example In the second example, we analyze the behavior of the algorithm when the hierarchy in the state equations is not satisfied. To this end, we modify the probability distributions that govern the state evolution as

$$\begin{aligned} p(\theta_{1,t} \mid \theta_{1,t-1}) &\sim \mathcal{N}(\theta_{1,t}; 1 + \theta_{1,t-1}, \sigma_{\theta_1}^2) \\ p(\theta_{2,t} \mid \theta_{2,t-1}, \theta_{1,t}) &\sim \mathcal{N}(\theta_{2,t}; \theta_{2,t-1}, \sigma_{\theta_2}^2) \\ p(\theta_{3,t} \mid \theta_{3,t-1}, \theta_{2,t}, \theta_{1,t}) &\sim \mathcal{N}(\theta_{3,t}; \theta_{3,t-1}, \sigma_{\theta_3}^2). \end{aligned} \tag{4.21}$$

The probability distributions corresponding to the three measurements were the same as the the distributions that were used in the first example, and the variances corresponding to the measurement noise and the process noise were the same as the ones used in the first example. In Fig. 4.2, we plot the RMSE for all the methods for the second example. It can be seen that the RMSE in the estimates of the partitions using the HPF was lower than the RMSE obtained using SPF-IO, and SPF-IL, but it was similar to the RMSE obtained using the SPF-LO.

4.6 Multiple-Target Tracking using HPF

In this section, we apply the proposed algorithm for the problem of multiple target tracking. In this chapter, we address a tracking scenario where the number of targets is

unknown and the tracking system should be capable of automatically determining this number. We consider a network comprising three kinds of sensors: a multistatic radar with one transmit and three receive antennas, which can measure the backscattered signal from the targets; one infrared camera, which can obtain the top view of the region using arial shots; and an intelligence report provided by a human scout.

4.6.1 System Model

We consider a planar region of the battlefield, $\mathcal{R}$, with an unknown number of moving targets of various types. At time t, we assume that the number of targets is N_t. Note that we changed the notation here and we are using N_t to represent the number of the targets at time t. The targets are indexed as $\{1, \ldots, n_t, \ldots, N_t\}$ with the position and velocity of n_t^{th} target denoted as $\boldsymbol{\rho}_{n_t} = [x_{n_t}, y_{n_t}]^T \in \mathbb{R}^2$ and $\dot{\boldsymbol{\rho}}_{n_t} = [\dot{x}_{n_t}, \dot{y}_{n_t}]^T \in \mathbb{R}^2$, respectively. We label the target categories using $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_{n_t}, \ldots, \alpha_{N_t}\}$, where $\alpha_{n_t} \in \mathcal{A}$, with $\mathcal{A}$ being the set of all the target types. The parameters of interest are the number of targets N_t in the scene, the positions $\boldsymbol{\rho}_{n_t}$, the velocities $\dot{\boldsymbol{\rho}}_{n_t}$, and the categories α_{n_t} of each of the targets. The overall state vector at time t is obtained by concatenating all the unknown parameters and we denote it using $\boldsymbol{\theta}_t = [N_t, \boldsymbol{\rho}_1^T, \dot{\boldsymbol{\rho}}_1^T, \alpha_1, \cdots, \boldsymbol{\rho}_{N_t}^T, \dot{\boldsymbol{\rho}}_{N_t}^T, \alpha_{N_t}]^T \in (\mathbb{R}^2 \times \mathbb{R}^2 \times \mathcal{A})^{N_t}$. Next, we derive the state transition model for the vector $\boldsymbol{\theta}_t$.

4.6.2 State-Space Model

We assume, for simplicity, that there can be at most one birth or one death of the targets at each state transition, and we represent the probabilities of the death and

the birth of the targets using p_{d} and p_{b}, respectively. Hence, we have

$$p(N_{t+1}|N_t = n_t) = \begin{cases} p_{\mathrm{b}} & \text{if } N_{t+1} = n_t + 1, \\ n_t p_{\mathrm{d}}(1-p_{\mathrm{d}})^{n_t-1} & \text{if } N_{t+1} = n_t - 1, \\ 1 - p_{\mathrm{b}} - n_t p_{\mathrm{d}}(1-p_{\mathrm{d}})^{n_t-1} & \text{if } N_{t+1} = n_t. \end{cases} \tag{4.22}$$

Let $\boldsymbol{\alpha}_t = \{\alpha_{t,1}, \alpha_{t,2}, \cdots, \alpha_{t,N_t}\}$ denote the categories of each of the N_t targets, and assume that the number of possible categories is finite, i.e., $\mathrm{card}(\mathcal{A}) = \mathrm{M} < \infty$. Let $\alpha^* \in \mathcal{A}$ be the category of the new target that appears at time $t+1$. The probability distribution for the target categories $\boldsymbol{\alpha}_{t+1}$ at time $t+1$ given $\boldsymbol{\alpha}_t$ and N_{t+1} can be written as

$$p(\boldsymbol{\alpha}_{t+1}|\boldsymbol{\alpha}_t, N_{t+1}) = \begin{cases} \frac{1}{M} & \text{if } N_{t+1} = N_t + 1, \boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t \cup \alpha^*, \\ \frac{1}{N_t} & \text{if } N_{t+1} = N_t - 1, \boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t - \alpha_{t,n_t} \text{ for any } n_t, \\ \frac{p_{\mathrm{d}} p_{\mathrm{b}}}{N_t M} & \text{if } N_{t+1} = N_t, \boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t - \alpha_{t,n_t} \cup \alpha^*, \\ 1 - \frac{p_{\mathrm{d}} p_{\mathrm{b}}}{N_t M} & \text{if } N_{t+1} = N_t, \boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t. \end{cases} \tag{4.23}$$

Let $\boldsymbol{\chi}_t$ denote the set of all the targets present in the scene at time t. We now define the state transitions of the targets that are present at both times t and $t+1$. For such a target, $n_t \in \{\boldsymbol{\chi}_{t+1} \cap \boldsymbol{\chi}_t\}$, we define a vector of its position and velocity as $\boldsymbol{\xi}_{t+1,n_t} = [\boldsymbol{\rho}^T_{t+1,n_t}, \dot{\boldsymbol{\rho}}^T_{t+1,n_t}]^T$. Then, given $\boldsymbol{\xi}_{t,n_t}$, we have

$$\boldsymbol{\xi}_{t+1,n_t} = \boldsymbol{F}_{n_t} \boldsymbol{\xi}_{t,n_t} + \boldsymbol{v}_{t,n_t}, \tag{4.24}$$

where $\boldsymbol{F}_{n_t}$ is the state transition matrix and $\boldsymbol{v}_{t,n_t}$ is the process noise. We assume that the targets follow linear trajectories and the process noise, $\boldsymbol{v}_{t,n_t}$, is Gaussian

distributed with a zero mean and a covariance matrix $\mathbf{\Sigma}_{\mathrm{v},n_t}$. Hence, we have

$$p(\boldsymbol{\xi}_{t+1,n_t}|\boldsymbol{\xi}_{t,n_t}) = \mathcal{N}(\boldsymbol{\xi}_{t+1,n_t}; \boldsymbol{F}_{n_t}\boldsymbol{\xi}_{t,n_t}, \mathbf{\Sigma}_{\mathrm{v},n_t}), \tag{4.25}$$

The state transition of the vector $\boldsymbol{\theta}_t$ can be obtained by using the chain rule as

$$p(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t) = p(\boldsymbol{\xi}^*_{t+1})p(\boldsymbol{\alpha}_{t+1}|\boldsymbol{\alpha}_t, N_{t+1})p(N_{t+1}|N_t) \prod_{n_t \in \boldsymbol{\chi}_{t+1} \cap \boldsymbol{\chi}_t} p(\boldsymbol{\xi}_{t+1,n_t}|\boldsymbol{\xi}_{t,n_t}) \tag{4.26}$$

where $p(\boldsymbol{\xi}^*_{t+1})$ is the probability density of the position and velocity vector of the new target initiated at time $t+1$. We assume that targets belonging to any of the categories can enter the scene with an equal probability. Further, for simplicity, we assume that all the targets belonging to a particular category will enter the scene at the same location.

4.6.3 Measurement Models

In this subsection, we describe the sensor measurement models for the multistatic radar, the infrared camera and the human scout. At time t, let $\boldsymbol{y}_t = \{\boldsymbol{y}_{1,t}, \boldsymbol{y}_{2,t}, \boldsymbol{y}_{3,t}\}$ be the measurements obtained from the sensors. $\boldsymbol{y}_{1,t}$ corresponds to the measurement obtained from a multistatic radar system, $\boldsymbol{y}_{2,t}$ corresponds to the measurement obtained from an infrared camera, and $\boldsymbol{y}_{3,t}$ corresponds to the measurement obtained by the human scout.

Multistatic Radar: As before, we assume that the transmit antenna uses coherent train of multiple pulses with a pulse repetition period of t_{p} seconds:

$$s(t') = \sum_{l=0}^{L-1} a_l(t' - lt_{\mathrm{p}}), \tag{4.27}$$

where $a_l(t')$ is the transmitted signal in the l^{th} pulse. The discrete version of the signal $a_l(t')$ is assumed to be of length G and it is denoted as $\boldsymbol{a}_l$. Let τ_{p,n_t} be the total time taken for the signal to travel from the transmit antenna to the n_t^{th} target and back to the p^{th} receive antenna, and ν_{p,n_t} be the Doppler frequency shift due the n_t^{th} target. The received signal at the p^{th} receive antenna, due to the signal bouncing off the n_t^{th} target, can then be expressed as

$$y_{1,p}(t') = \beta_{p,n_t}\sqrt{\gamma_1}s(t' - \tau_{p,n_t})e^{j2\pi\nu_{p,n_t}t'} + w_{1,p}(t'), \tag{4.28}$$

where β_{p,n_t} is the target RCS which is a function of the target category, γ_1 is the transmit signal energy, and $w_{1,p}(t')$ is the additive receiver noise. We assume that the mapping from the target category to the target RCS is known and one-to-one. The noise is assumed to be circularly symmetric, complex, white, and following a Gaussian distribution. The corresponding discrete-time signal can be obtained by sampling the received signal and considering only N_{p} samples. It can be expressed as

$$\boldsymbol{y}_{1,p,n_t} = \beta_{p,n_t}\sqrt{\gamma_1}\underbrace{\Big(\boldsymbol{\Upsilon}(p,n_t)\otimes\boldsymbol{\Gamma}(p,n_t)\Big)}_{\boldsymbol{\phi}_{p,n_t}}\boldsymbol{s} + \boldsymbol{w}_{1,p}, \tag{4.29}$$

where

- $\boldsymbol{y}_{1,p,n_t}$ is a $LN_{\mathrm{p}} \times 1$ received signal vector at the p^{th} antenna due to the signal bouncing off the n_t^{th} target
- $\boldsymbol{\Upsilon}(p,n_t)$ is an $L \times L$ Doppler modulation matrix defined as $\mathrm{diag}\big\{1, e^{j2\pi\nu_{p,n_t}t_{\mathrm{p}}}, \cdots, e^{j2\pi\nu_{p,n_t}(L-1)t_{\mathrm{p}}}\big\}$
- $\boldsymbol{\Gamma}(p,n_t)$ is a $N_{\mathrm{p}}\times G$ time shift matrix defined as $\Big[\ \boldsymbol{0}_{\tilde{\tau}_{p,n_t}\times G};\quad \boldsymbol{I}_G;\quad \boldsymbol{0}_{N_{\mathrm{s}}-G-\tilde{\tau}_{p,n_t}\times G}\ \Big]$

- $\boldsymbol{s}$ is a $LG \times 1$ column vector obtained by stacking the transmitted signal in each pulse i.e., $\boldsymbol{s} = [\boldsymbol{a}_0^T, \boldsymbol{a}_1^T, \ldots, \boldsymbol{a}_{L-1}^T]^T$
- $\boldsymbol{w}_{1,p}$ is a $LN_{\mathrm{s}} \times 1$ complex additive white Gaussian noise at the p^{th} receiver with a zero mean and a covariance matrix $\boldsymbol{\Sigma}_{1,p} = \sigma_{1,p}^2 \boldsymbol{I}_{LN_{\mathrm{p}}}$

The received signal due to all the targets can now be expressed in a matrix form as

$$\boldsymbol{y}_{1,p,t} = \sum_{n_t=1}^{N_t} \beta_{p,n_t} \underbrace{\sqrt{\gamma_1}\Big(\boldsymbol{\Upsilon}(p,n_t) \otimes \boldsymbol{\Gamma}(p,n_t)\Big)\boldsymbol{s}}_{\boldsymbol{\phi}_{p,n_t}} + \boldsymbol{w}_{1,p} = \boldsymbol{\Phi}_{p,t}\boldsymbol{\beta}_{p,t} + \boldsymbol{w}_{1,p}, \tag{4.30}$$

where

- $\boldsymbol{y}_{1,p,t}$ is a $LN_{\mathrm{p}} \times 1$ received signal vector at the p^{th} antenna due to all the targets
- $\boldsymbol{\Phi}_{p,t}$ is a $LN_{\mathrm{p}} \times N_t$ matrix defined as $\boldsymbol{\Phi}_{p,t} = [\boldsymbol{\phi}_{p,1}, \ldots, \boldsymbol{\phi}_{p,N_t}]$
- $\boldsymbol{\beta}_{p,t}$ is a $N_t \times 1$ vector defined as $\boldsymbol{\beta}_t = [\beta_{p,1}, \ldots, \beta_{p,N_t}]$

Hence we have

$$p(\boldsymbol{y}_{1,p,t}|\boldsymbol{\theta}_t) = \mathbb{C}\mathcal{N}(\boldsymbol{y}_{1,p,t}; \boldsymbol{\Phi}_{p,t}\boldsymbol{\beta}_{p,t}, \boldsymbol{\Sigma}_{1,p}), \quad \text{for } p = 1, 2, 3. \tag{4.31}$$

Since the measurements obtained at the receive antennas are independent of each other, we have

$$p(\boldsymbol{y}_{1,t}|\boldsymbol{\theta}_t) = \mathbb{C}\mathcal{N}(\boldsymbol{y}_{1,t}; \boldsymbol{\Phi}_t\boldsymbol{\beta}_t, \boldsymbol{\Sigma}_1), \tag{4.32}$$

where $\boldsymbol{y}_{1,t} = [\boldsymbol{y}_{1,1,t}^T, \boldsymbol{y}_{1,2,t}^T, \boldsymbol{y}_{1,3,t}^T]^T$, $\boldsymbol{\Phi}_t = \mathrm{blkdiag}\{\boldsymbol{\Phi}_{1,t}, \boldsymbol{\Phi}_{2,t}, \boldsymbol{\Phi}_{3,t}\}$, $\boldsymbol{\beta}_t = [\boldsymbol{\beta}_{1,t}^T, \boldsymbol{\beta}_{2,t}^T, \boldsymbol{\beta}_{3,t}^T]^T$ and $\boldsymbol{\Sigma}_1 = \mathrm{blkdiag}\{\boldsymbol{\Sigma}_{1,1}, \boldsymbol{\Sigma}_{1,2}, \boldsymbol{\Sigma}_{1,3}\}$.

Infrared Camera: We use a measurement model similar to the one proposed in [80] and [81] for the infrared camera. The output of an infrared camera, which is an $R \times C$ matrix of pixel values, is modeled as a noisy version of an ideal image $\boldsymbol{I}_0$ convolved with the point-spread function of the camera. For simplicity, we assume the point-spread function to be a delta function and the ideal image to be of the form

$$\boldsymbol{I}_0(\boldsymbol{z}) \;=\; \sum_{n_t=1}^{N_t} T_{n_t}\delta(\boldsymbol{z}-\boldsymbol{\rho}_{n_t}), \tag{4.33}$$

where $\boldsymbol{z}$ is a two-dimensional pixel location in the image obtained by the camera, $\boldsymbol{\rho}_{n_t}$ is the two-dimensional pixel location of the n_t^{th} target in the image obtained by the camera, which depends on the actual location of the n_t^{th} target, and T_{n_t} is a constant that depends on the target category. Following [80], we express the likelihood of the output of the camera as

$$p(\boldsymbol{y}_{2,t} \mid \boldsymbol{\theta}_t) = \mathcal{N}(\boldsymbol{y}_{2,t}; \mathrm{vec}(\gamma_2 \boldsymbol{I}_0), \boldsymbol{\Sigma}_2), \tag{4.34}$$

where $\boldsymbol{y}_{2,t}$ is a vector form of the matrix of pixel values that correspond to the output of the camera, γ_2 is a constant that depends on the quality of the camera, and $\boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_{2,1} \otimes \boldsymbol{\Sigma}_{2,2}$ where $\boldsymbol{\Sigma}_{2,1}$ and $\boldsymbol{\Sigma}_{2,2}$ are the covariance matrices of the measurement noise along the rows and the columns, respectively.

Human scout: The measurement report given by the scout is an M-dimensional vector with its m^{th} entry representing the number of targets of type m. The total number of targets at time t, as counted by the scout, is given as

$$N_{3,t} = \sum_{m=1}^{M} y_{3,t,m}, \tag{4.35}$$

where $y_{3,t,m}$ is the m^{th} entry of the vector $\boldsymbol{y}_{3,t}$. In the above, we use a subscript, $N_{3,t}$, to denote the number of targets counted by the scout in order to differentiate it from the actual number of targets in the scene which is denoted by N_t. Let $N_{\max}$ be an upper bound on $N_{3,t}$. We obtain the probability mass distribution for the number of targets counted by scout, $N_{3,t}$, denoted $g(N_{3,t})$, by evaluating a Gaussian density [8] with mean N_t, the actual number of targets, and variance $\sigma^2_{3,t}$ followed by normalizing:

$$g(N_{3,t} = k) = \frac{\mathcal{N}(k; N_t, \sigma^2_{3,t})}{\sum_{i=1}^{N_{\max}} \mathcal{N}(i; N_t, \sigma^2_{3,t})}, N_t > 0 \tag{4.36}$$

The variance $\sigma^2_{3,t}$ is chosen to be N_t/γ_3, where the parameter γ_3 is proportional to the level of the training that the scout undergoes and the quality of the equipment that he uses. The variance also depends on the actual number of the targets in the scene. The greater the number of targets, the higher the probability that the scout incorrectly counts them. A similar probability mass function was used in [81] for the number of targets counted by a human scout. Let p_{c} denote the probability that the scout counts at least one target incorrectly. We model the probability p_{c} to be inversely proportional to γ_3, i.e., $p_{\mathrm{c}} = b/\gamma_3$, where b is a known constant chosen such that $p_{\mathrm{c}} \in [0, 1]$. From the expression for p_{c}, it can be seen that well-trained scouts and scouts with better equipment have a lower probability of incorrectly identifying the targets. We model the scout's measurements $\boldsymbol{y}_{3,t}$ to follow a multinomial distribution, whenever at least one target is identified incorrectly. The likelihood for the scout's measurement report can then be expressed as

$$p(\boldsymbol{y}_{3,t}|\boldsymbol{\theta}_t) = \begin{cases} p_{\mathrm{c}} \sum_{k=1}^{N_{\max}} g(k) \frac{k!}{y_{3,t,1}! \cdots y_{3,t,M}!} q_{1,t}^{y_{3,t,1}} \cdots q_{M,t}^{y_{3,t,M}} & \text{if at least one target is incorrectly identified,} \\ (1 - p_{\mathrm{c}}) + p_{\mathrm{c}}[g(N_{3,t}) \frac{N_{3,t}!}{y_{3,t,1}! \cdots y_{3,t,M}!} q_{1,t}^{y_{3,t,1}} \cdots q_{M,t}^{y_{3,t,M}}] & \text{otherwise.} \end{cases} \tag{4.37}$$

[8] Note here that we are computing a discrete probability distribution by sampling a continuous Gaussian density. The resulting probability mass function for the number of targets counted by the scout is still a discrete distribution.

The probabilities $q_{1,t}, q_{2,t}, \ldots, q_{M,t}$ are obtained based on the actual values of the state vector $\boldsymbol{\theta}_t$. We first evaluate an $N_t \times M$ matrix $\boldsymbol{Q}_t$, such that

$$[\boldsymbol{Q}_t]_{n_t,m} = \begin{cases} 1 - q_{\mathrm{c}} & \text{if the } n_t^{th} \text{ target is of type } m, \\ \frac{q_{\mathrm{c}}}{M-1} & \text{otherwise} \end{cases}, \tag{4.38}$$

where $0 \leq q_{\mathrm{c}} \leq 1$ is a known constant. We then normalize $\boldsymbol{Q}_t$ to obtain the probability vector $q_{m,t}$ as $q_{m,t} = \frac{1}{N_t} \sum_{n_t=1}^{N_t} [\boldsymbol{Q}_t]_{n_t,m}$. We assumed in this work that the samples from all the sensors arrive at the receiver at the same time and that the sensors are time-synchronized.

4.6.4 Filtering Algorithm

We divide the state-space into three partitions. The first partition $\boldsymbol{\theta}_{1,t}$ comprises the target number and the target categories, i.e., $\boldsymbol{\theta}_{1,t} = [N_t, \boldsymbol{\alpha}_t^T]^T$. The second and the third partitions comprise the target positions and velocities, respectively, i.e., $\boldsymbol{\theta}_{2,t} = [\boldsymbol{\rho}_{1,t}^T, \ldots, \boldsymbol{\rho}_{N_t,t}^T]^T$ and $\boldsymbol{\theta}_{3,t} = [\dot{\boldsymbol{\rho}}_{1,t}^T, \ldots, \dot{\boldsymbol{\rho}}_{N_t,t}^T]^T$. It can be seen that the evolution of the second and third partitions depend on the evolution of the first partition. The measurements collected by the scout depend only on the first partition, the measurements collected by the infrared camera are a function of both the first and the second partitions, and the measurements collected by the radar are function of all the three partitions. Thus the measurement and the state functions satisfy the hierarchical model that is described in Eq. (4.4). In order to obtain the state estimate, we use the HPF that is described in Sec. 4.5.

In general, it is observed that partitioning the samples of each target individually results in a better filtering performance [82], [83]. The proposed hierarchical filtering

can be extended to operate in this manner by further dividing the second and third partitions $\boldsymbol{\theta}_{2,t}$ and $\boldsymbol{\theta}_{3,t}$.

4.7 Numerical Results

In this section, we use numerical examples to demonstrate the performance of the tracking system when the proposed hierarchical filtering method is used for target tracking. In order to quantify the performance of the multiple target tracking system, we define four performance metrics. We describe the simulation setup first and then discuss the examples.

Target parameters: We consider surveillance of a region for a period of 20 tracking intervals. The duration of each tracking interval was 0.1 s ($\Delta t = 0.1s$). We consider tracking under two scenarios. In the first scenario, during the first 0.8 s, which corresponds to 8 intervals, there were 3 targets in the scene, during the next 0.8 s, i.e., between the 9^{th} and the 15^{th} interval there were 4 targets, and thereafter there were 3 targets again. The number of target classes was chosen to be 5 ($M = 5$), and the initial positions and initial velocities of the targets belonging to various classes were chosen as shown in the Table 4.1. During the entire tracking period, the target categories were chosen as

$$\boldsymbol{\alpha}_t = \begin{cases} \{1,2,4\} & t = 1,2,\ldots,8 \\ \{1,2,4,5\} & t = 9,10,\ldots,15 \\ \{1,2,5\} & t = 16,\ldots,20 \end{cases}$$

In the second scenario, there were three targets during the entire duration, i.e. $\boldsymbol{\alpha}_t = \{1, 2, 4\}$ for $t = 1, \ldots, 20$. The initial positions and velocities of these targets were again chosen according to Table 4.1. The probabilities of the birth and the death of the targets were chosen to be 0.01, i.e., $p_{\mathrm{d}} = p_{\mathrm{b}} = 0.01$, respectively. The constants T_{n_t} in the measurement model of the infrared camera (see Eq. (4.33)) were chosen to be $T_1 = 4.2, T_2 = 8, T_3 = 9, T_4 = 10.4, T_5 = 13.6$, respectively.

Signal and Sensor Parameters: In each pulse, we transmit a spread-spectrum waveform [40]- [39] with 16 ($G = 16$) chips from the radar antenna. The total bandwidth was 100Mhz ($B = 100$) and the carrier frequency, f_{c}, of the transmitted waveforms was 1Ghz. We used four ($L = 4$) pulses in each tracking interval. The radar receive antennas were located at $(x_{\mathrm{Rx},1}, y_{\mathrm{Rx},1}) = (0, 0), (x_{\mathrm{Rx},2}, y_{\mathrm{Rx},2}) = (20, 0), (x_{\mathrm{Rx},3}, y_{\mathrm{Rx},3}) = (40, 0)$, respectively, and the variance of the measurement noise at each receiver was $\sigma_{1,p}^2 = 1 \times 10^{-3}, p = 1, 2, 3$. The covariance matrices of the measurement noise at the infrared camera were chosen to be $\boldsymbol{\Sigma}_{2,1} = \sigma_2^2 \boldsymbol{I}_R, \boldsymbol{\Sigma}_{2,2} = \sigma_2^2 \boldsymbol{I}_C$, with $\sigma_2^2 = 1 \times 10^{-2}$. The constants b and q_c for the human scout were chosen to be 0.2 and 0.05, respectively. We assumed that all the targets are observable by all the sensors. We evaluated the

Table 4.1: Table showing the initial positions and velocities of all the target classes

Target Class	Initial Position		Initial Velocity	
	Fixed	Varying	Fixed	Varying
1	(5,5)	(5,5)	(10,10)	(10,0)
2	(10,35)	(10,0)	(5,0)	(0,10)
3	(20,20)	(20,20)	(3,3)	(3,3)
4	(35,35)	(12,15)	(-10,-10)	(0,-10)
5	(0,30)	(8,12)	(5,-10)	(10,-5)

performance of the system using four metrics: the average number of targets detected in the scene, the average number of targets identified incorrectly, the root mean-squared error in the position of correctly identified targets, and the root mean-squared

error in the velocity of correctly identified targets. Let the estimate of the state vector at time t in the i^{th} Monte-Carlo iteration be $\hat{\boldsymbol{\theta}}_{t,i} = [\hat{N}_{\mathrm{t}}, \hat{\boldsymbol{\rho}}_1^T, \hat{\dot{\boldsymbol{\rho}}}_1^T, \hat{\alpha}_1, \cdots, \hat{\boldsymbol{\rho}}_{N_t}^T, \hat{\dot{\boldsymbol{\rho}}}_{N_t}^T, \hat{\alpha}_{N_t}]_i^T$. The four performance metrics are then defined as:

$$
\begin{aligned}
\mathrm{PM}_1 &= \frac{1}{N_{\mathrm{mc}}} \sum_{i=1}^{N_{\mathrm{mc}}} \hat{N}_{t,i}, \\
\mathrm{PM}_2 &= \frac{1}{N_{\mathrm{mc}}} \sum_{i=1}^{N_{\mathrm{mc}}} \mathrm{card}\Big((\boldsymbol{\alpha}_t - \hat{\boldsymbol{\alpha}}_{t,i}) \bigcup (\hat{\boldsymbol{\alpha}}_{t,i} - \boldsymbol{\alpha}_t) \Big), \\
\mathrm{PM}_3 &= \frac{1}{|B_t| N_{\mathrm{mc}}} \sum_{i=1}^{N_{\mathrm{mc}}} \sum_{n_t \in B_t} \sqrt{(\rho_{\mathrm{x},n_t} - \hat{\rho}_{\mathrm{x},n_t})^2 + (\rho_{\mathrm{y},n_t} - \hat{\rho}_{\mathrm{y},n_t})^2}, \\
\mathrm{PM}_4 &= \frac{1}{|B_t| N_{\mathrm{mc}}} \sum_{i=1}^{N_{\mathrm{mc}}} \sum_{n_t \in B_t} \sqrt{(\dot{\rho}_{\mathrm{x},n_t} - \hat{\dot{\rho}}_{\mathrm{x},n_t})^2 + (\dot{\rho}_{\mathrm{y},n_t} - \hat{\dot{\rho}}_{\mathrm{y},n_t})^2},
\end{aligned}
$$

where B_t is the set of correctly identified targets.

In Figs. 4.3, we plot these metrics using the four methods, for varying number of targets. In the first method, we used the proposed HPF. In the second, third, and fourth methods, we used SPF-LO, SPF-IO, SPF-IL, where the weights were updated following Eqs. (4.8), (4.10), and (4.12), respectively. We used $N_{\mathrm{s}} = 1000$ particles for the simulations and the results were averaged over $N_{\mathrm{mc}} = 50$ Monte Carlo iterations.

In Fig. 4.3(a), we plot the actual number of targets (ground truth), labeled as GT along with estimates of the number of targets obtained using other methods. It can be seen from this figure that the system was able to accurately estimate the number of targets using the HPF approach. On the other hand, using SPF-LO, SPF-IO, and SPF-IL resulted in an incorrect estimation of the number of targets. It can also be seen that using HPF resulted in higher correct identifications of the target categories compared to the other methods. However, from Fig. 4.3(d), it can be seen that the

RMSE in the range and velocity, per target, for the correctly identified targets, using HPF was comparable to the RMSE obtained using the SPF based methods.

In Fig. 4.4, we plot the performance metrics for a fixed number of targets. It can be seen that using HPF resulted in an accurate estimate of the target number, higher correct identifications, and lower RMSE in range when compared to the performance obtained using SPF based methods. Among the other methods, SPF-IL performed better, followed by SPF-LO and SPF-IO. The RMSE in the velocity estimate using HPF was comparable to the RMSE using SPF based methods. This is due to the fact that the velocity and the range evolve independent of each other. We plot the actual target trajectories for the two scenarios and the estimated trajectories (using a single Monte-Carlo run) using the proposed HPF in Fig. 4.5.

In Table 4.2, we list the average computational time taken by each of these algorithms for the case of varying numbers of targets. We considered the average running time for the case of $N_{\mathrm{s}} = 500, 1000$ and 2000 particles. The simulations were run on an 8-core processor using parallel matlab sessions. HPF requires a sampling and a weight update step in each stage of the filtering process, and as a result it takes longer time compared to other SPF based methods. An $n-$stage HPF requires n sample generation and weight update steps, whereas an SPF based method requires only one sampling and weight update step.

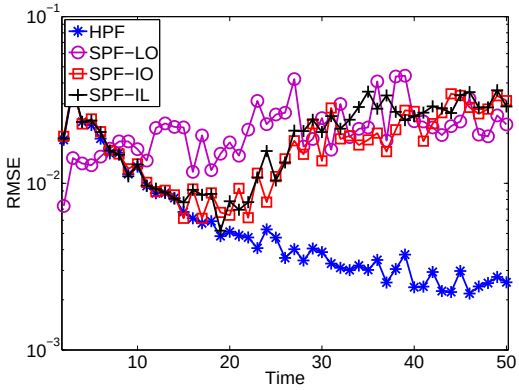
Table 4.2: Table showing average computational time in seconds for various algorithms

No. of Particles	HPF	SPF-LO	SPF-IO	SPF-IL
500	4.23	1.02	0.98	0.89
1000	9.36	2.26	1.96	1.85
2000	18.76	4.19	3.74	2.98

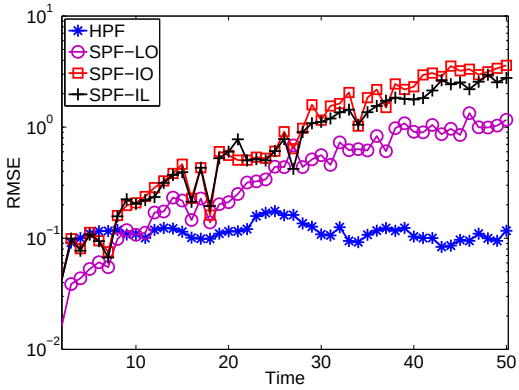
4.8 Summary

In this chapter, we proposed a new filter, called the hierarchical particle filter (HPF) to obtain the state estimate. HPF finds the estimate of the state vector in several stages. At each stage, the filter uses the posterior weights obtained in the last stages as extrinsic information, and computes the estimate of the state vector in the current stage. The data obtained from multiple sensing modalities is combined in both the time-update and the measurement-update step of the recursive Bayesian filtering.

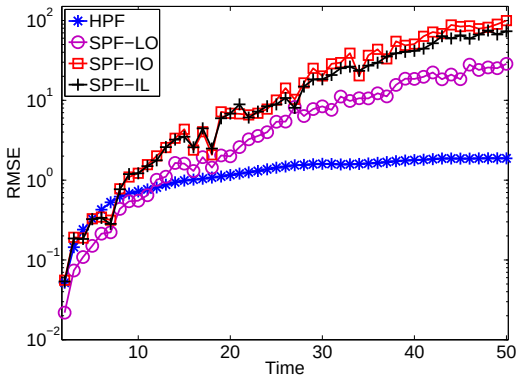
We then used the proposed filtering method for the joint initiation, termination and tracking of multiple targets using multi-modal sensors. We used a sensor network comprising three different types of sensors: a radar, an infrared camera and a human scout. We compared the tracking performance with standard particle filtering (SPF) using linear opinion (SPF-LO), independent opinion (SPF-IO) and independent likelihood (SPF-IL) for the data fusion. The results demonstrated that HPF was able to accurately identify the number of targets, produce higher correct identifications, and lower root mean-squared error (RMSE) in range estimates when compared to the performance obtained using the SPF-LO, SPF-IO, and SPF-IL. The RMSE in the velocity estimates was similar for all the methods.



(a) RMSE - θ_1

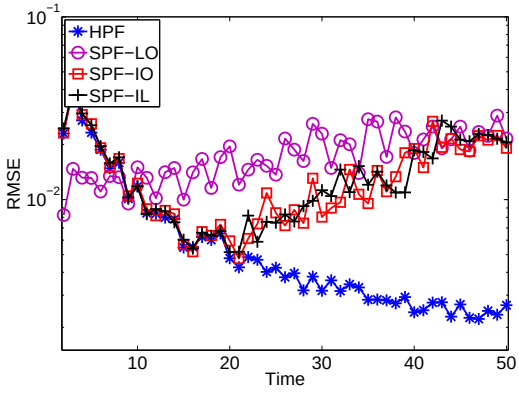


(b) RMSE - θ_2

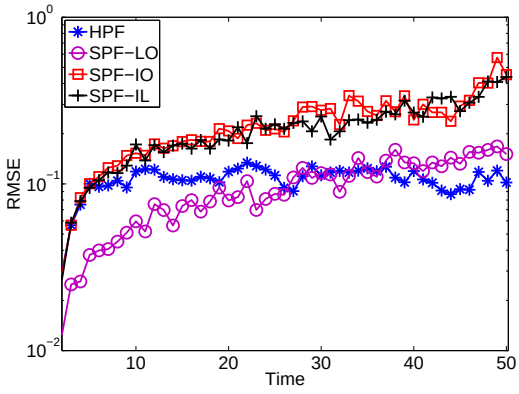


(c) RMSE - θ_3

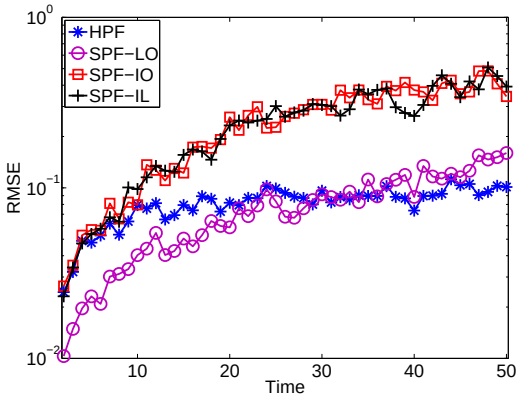
Figure 4.1: RMSE for the three partitions for the first example.



(a) RMSE - θ_1

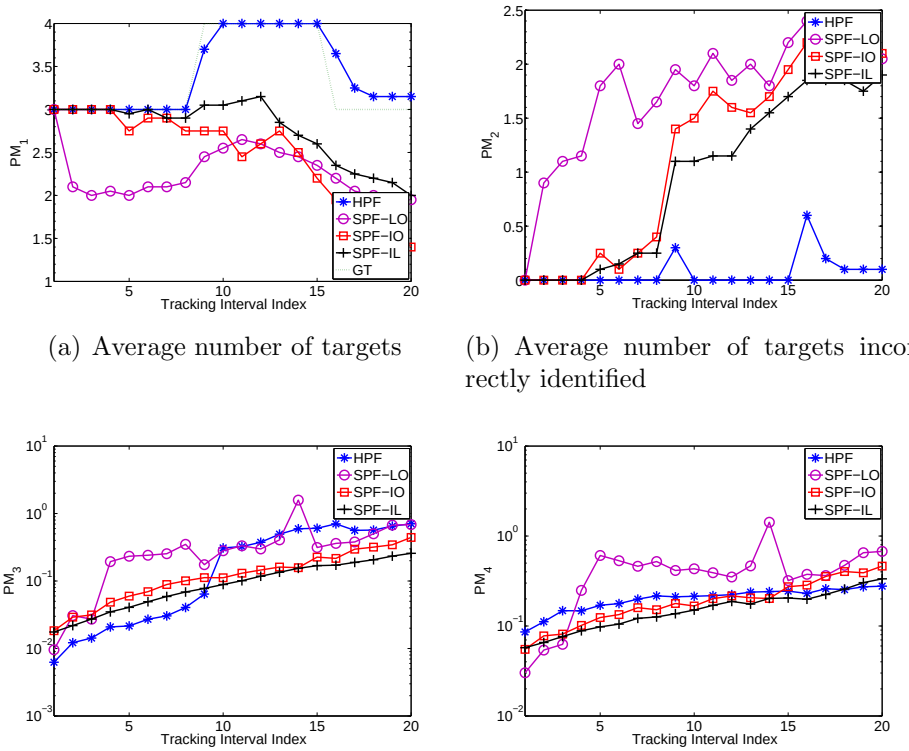


(b) RMSE - θ_2



(c) RMSE - θ_3

Figure 4.2: RMSE for the three partitions for the second example.



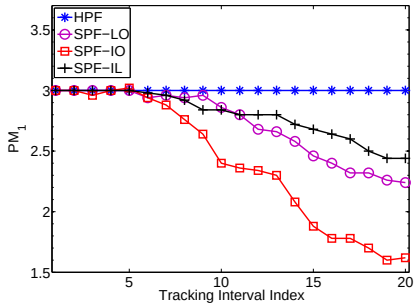
(a) Average number of targets

(b) Average number of targets incorrectly identified

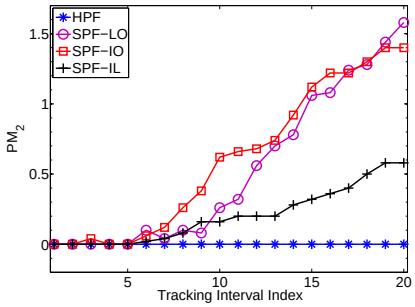
(c) RMSE in range

(d) RMSE in velocity

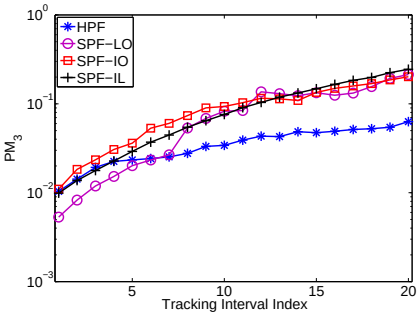
Figure 4.3: Performance comparison for varying number of targets.



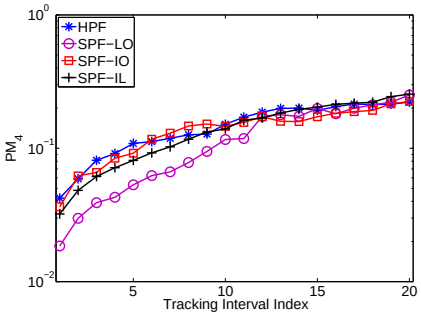
(a) Average number of targets



(b) Average number of targets incorrectly identified

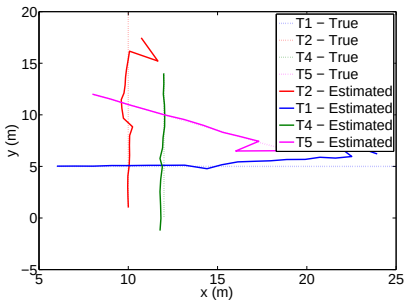


(c) RMSE in range

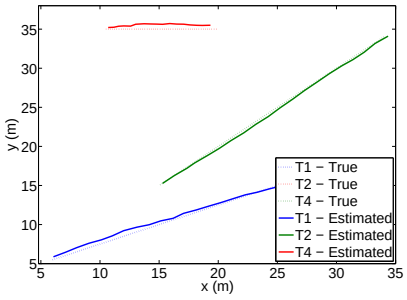


(d) RMSE in velocity

Figure 4.4: Performance comparison for fixed number of targets.



(a) Scenario 1



(b) Scenario 2

Figure 4.5: Estimated Vs Actual target trajectories.

Chapter 5

Multi-Modal Sensor Management using Price Theory[9]

5.1 Introduction

As described in the previous chapter, the advances in the development of inexpensive sensing devices has made it feasible to deploy a multi-modal sensor network to obtain information about various targets, activities and events. However, typically, the multi-modal data collected across the sensors is highly diverse, incomparable, and sometimes inconsistent. Therefore, the sensing system should understand how to combine the information provided by multiple sensors, and it should be capable of adaptively changing the way it combines this information depending on the situation. In addition, gathering information using the sensors incurs cost: physical cost, computational cost, and communication cost. Sensing applications, however, are constrained by limited resources. Hence, the sensing system should decide which sensors it should use to produce the data, and how it should distribute the limited

[9]Based on P. Chavali and A. Nehorai, "Managing Multi-Modal Sensor Networks using Price Theory", *IEEE Trans. on Signal Processing*, vol. 60, no. 9, pp. 4874-4887, Sep. 2012. ©[2012] IEEE

resources to the sensors, such that the system achieves the best performance within the cost constraints. These considerations give rise to three important aspects in sensor management: sensor selection (SS) [84], resource allocation (RA) and data fusion (DF) [85]. Sensor management can thus be defined as a process that manages and coordinates the use of sensors by adaptively selecting the sensors, distributing the resources among the sensors, and combining the information obtained from various sensors with an overall goal of improving the system performance.

In this chapter, we propose a framework for sensor management that is inspired by the trading behavior of agents in a commercial market and we use an economic price theory-based approach [86]. The idea of a using a market-based approach is not new [87], and it has been used to solve the resource allocation problems in the past. In our work, however, we use this approach for the entire sensor management process, by jointly addressing SS, RA, and DF. We model both the sensor nodes and the sensor manager as self interested agents in a double-sided market. This kind of modeling enables us to obtain a joint solution to all three problems related to sensor management. We model each sensor node (SN) as a producer who wishes to sell the data it produces, and each sensor manager (SM) as a consumer who wants to buy the data from the sensors. The basic idea is that the data is priced by looking to balance global supply vs global demand. The price of the data forces the SM to restrict itself to using the data only from a limited number of sensors. Since the SM must provide a payment for the data it uses, it will be straightforward in describing the utility it obtains from each kind of SN. In a similar way, the SN will purchase resources, for example, power, from the SM and pay in return for the resources. The SN will use these resources to produce the data and it will be straightforward in describing the quantity of the data it can provide with the resources it has. In this way, the proposed

approach will provide a natural framework for the SN and the SM to interact and to communicate their utilities dynamically. It also provides a rigorous methodology for incorporating the importance of the information that the sensors collect to the overall system goal. We describe the proposed framework using multiple-target tracking as an illustrative example.

5.2 Problem Description

In this section, we describe in detail the problem formulation. Let $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ denote the state vector. Consider a sensor network with P sensors, possibly of different modalities, labeled as $\mathbb{S} = \{1, 2, \ldots, P\}$. At time t, each of these sensors can partially observe the vector $\boldsymbol{\theta}_t$, i.e., the measured data $\boldsymbol{y}_{p,t}$ corresponding to the p^{th} sensor is a function of a subset $\boldsymbol{\theta}_{p,t} \subset \boldsymbol{\theta}_t$ of the state vector, which is represented as a projection defined on $\boldsymbol{\Theta}$. For the p^{th} sensor, we label this projection as $g_p(\boldsymbol{\theta}_p)$. The measurement vector corresponding to the p^{th} sensor at time t can then be written as

$$\boldsymbol{y}_{p,t} = g_p(\boldsymbol{\theta}_{p,t}; t) + \boldsymbol{w}_{p,t}, \tag{5.1}$$

where $\boldsymbol{w}_{p,t}$ denotes the additive noise. The function $g_p(\cdot)$ depends on the kind of the sensor used, and it is assumed to be known but possibly nonlinear. Let $\boldsymbol{Y}_t = \{\boldsymbol{y}_{1,t}, \ldots, \boldsymbol{y}_{P,t}\}$ denote the measurements obtained by all the sensors at time t and $\boldsymbol{y}_{p,1:t}$ be the measurements obtained by the p^{th} sensor up to time t. We assume the existence of a central fusion center, which we refer to as the sensor manager (SM) that processes the measurements obtained by the individual sensors and computes the global posterior distribution $p(\boldsymbol{\theta}_t \mid \boldsymbol{Y}_{1:t})$. The SM also decides which sensors should

be activated at each time, how the finite resources should be distributed among the activated sensors, and how the information obtained from each sensor should be combined. These considerations give rise to the three aspects of the sensor management: SS, RA and DF.

Let $c_{p,t}$ denote the the cost incurred by activating the p^{th} sensor at time t, $\pi_t \subset \mathbb{S}$ denote a subset of sensors at time t, $\mathcal{P}(\mathbb{S})$ denote the power set of $\mathbb{S}$, $\{u_t^i\}_{i=1}^3$ denote a set of utility functions that characterizes the system performance at time t, and $\boldsymbol{r}_{p,t}$ denote the resources allocated to the p^{th} sensor at time t. Using these notations, we define SS, RA, and DF as follows.

The problem of SS is to select a subset of sensors from a given set of possible sensors, with the constraints on the total cost and coverage. It can be formulated as the following constrained optimization problem:

$$\begin{aligned} \text{SS}: \quad & \pi_t^* = \arg\max_{\pi_t \in \mathcal{P}(\mathbb{S})} u_t^1, \\ \text{subject to } & \bigcup_{p \in \pi_t} \boldsymbol{\theta}_{p,t} = \boldsymbol{\theta} \text{ and } \sum_{p \in \pi_t} c_{p,t} \leq \eta_{\text{c}}. \end{aligned} \tag{5.2}$$

In this formulation, the first constraint corresponds to the coverage constraint, which ensures that every component of the state vector is covered by at least one sensor, and the second constraint ensures that the overall cost incurred by employing the sensors is within a predefined threshold. The parameter η_{c} corresponds to the constraint on the cost.

The problem of RA concerns allocating the available and limited resources to the sensors to maximize the utility each of them provides to the sensor network. It can

be formulated as the following joint constrained optimization problem:

$$\begin{aligned} \text{RA}: \quad \{\boldsymbol{r}^*_{p,t}\}_{p=1}^{P} &= \arg\max_{\{\boldsymbol{r}_{p,t}\}_{p=1}^{P}} u_t^2, \\ &\text{subject to } \sum_{p\in\pi_t} \boldsymbol{r}_{p,t} \leq \boldsymbol{\eta}_{\text{r}}. \end{aligned} \tag{5.3}$$

The parameter $\boldsymbol{\eta}_{\text{r}}$ corresponds to the total available resources.

The problem of DF is to find an optimal way to combine the information that is obtained from various sensors. Since the data collected by the sensors is, in general, inconsistent, it should be combined at the decision level [88]. In this chapter, we use the linear opinion pool [75] for the data fusion, as it is simple and widely used in practice. The problem of the DF then corresponds to finding the optimal weights, and it can be formulated as the following constrained optimization problem:

$$\begin{aligned} \text{DF}: \quad \boldsymbol{\mu}^*_t &= \arg\max_{\boldsymbol{\mu}_t\in\mathbb{R}^P} u_t^3 \ , \\ &\text{subject to } \sum_{p\in\pi^*_t} \mu_{p,t} = 1, \text{ and} \\ p(\boldsymbol{\theta}_t \mid \boldsymbol{Y}_{1:t}) &= \sum_{p\in\pi^*_t} \mu_p p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{p,1:t}), \end{aligned} \tag{5.4}$$

where the weight $\mu_{p,t}$ represents a subjective measure of the reliability of the data collected by the p^{th} sensor, and the utility function u_t^3 is a function of the global posterior distribution $p(\boldsymbol{\theta}_t \mid \boldsymbol{Y}_{1:t})$.

SS, RA, and DF have been studied independently in the past, and several techniques using various cost and utility functions, such as the reduction in the uncertainty of the state vector, volume of the confidence ellipsoid of the state vector etc., have been used. SS has been addressed in [58], [89], [90], [91], [92], RA has been studied in [93], [94] and DF in [72]. The list is not exhaustive. All these earlier methods

neglect the importance of dynamically changing environment on the utility functions that each SN and SM use. Further, there was no unified framework for obtaining a joint solution to SS, RA, and DF at the same time.

Our approach addresses these concerns by providing a framework, where the utility function of the SNs and the SMs is updated periodically. Each node of the sensor network acts as a self-interested economic agent that operates in a virtual market. A global efficient behavior is enforced by adjusting the price vector, which decides how each node behaves. The equilibrium point of this virtual market is then related to the solutions of SS, RA, and DF.

5.3 Price theory and Auctions: Preliminaries

5.3.1 Walrasian Equilibrium

Price theory is a branch of economics that explains the trade of goods and services between different economic agents [86]. Agents fall into two different categories: consumers and producers. Consumers can buy and sell various goods in the market, whereas producers can transform goods of some sort into goods of a different sort. Consider an economic market with N_{c} consumers, N_{p} producers, and K indivisible goods. For the i^{th} consumer, his preference for consuming various bundles of goods, denoted as $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iK}]^T$ and referred to as the demand vector, with $x_{ik} \in \mathbb{N}$ representing the quantity of k^{th} good that the i^{th} consumer trades, is specified by a utility function, $u_i : \mathbb{N}^K \rightarrow \mathbb{R}$. The utility function of a consumer ranks various bundles of goods according to his preference. If $x_{ik} > 0$, then the consumer buys the good, and if $x_{ik} < 0$, the consumer sells the good. Each consumer starts with

an initial endowment of goods $\boldsymbol{e}_i = [e_{i1}, \ldots, e_{iK}]^T$, with $e_{ik} \in \mathbb{N}$ representing the quantity of k^{th} good available for trade with the i^{th} consumer. Given a price vector $\boldsymbol{p} = [p_1, \ldots, p_K]^T$, the objective of the i^{th} consumer, $i = 1, \cdots, N_{\mathrm{c}}$, is to choose an optimal demand vector that maximizes his utility function under the constraint that the total wealth he spends is less than the total wealth he can generate by selling his endowment, given by $\sum_{k=1}^{K} e_{ik} p_k = \boldsymbol{p}^T \boldsymbol{e}_i$, at price $\boldsymbol{p}$. This feasible set is called the budget set of the consumer. The consumer's choice for his preferred bundle of goods is obtained by solving the following constrained optimization problem:

$$\begin{aligned} &\boldsymbol{x}_i^* = \arg \max_{\boldsymbol{x}_i \in B_i(\boldsymbol{p}, \boldsymbol{e}_i)} u_i(\boldsymbol{x}_i), \\ &\text{where } B_i(\boldsymbol{p}, \boldsymbol{e}_i) = \{\boldsymbol{x}_i \in \mathbb{N}^K : \boldsymbol{p}^T \boldsymbol{x}_i \leq \boldsymbol{p}^T \boldsymbol{e}_i\}, \quad i = 1, \cdots, N_{\mathrm{c}}. \end{aligned} \tag{5.5}$$

Agents of the second type, the producers, will take as input, goods from the consumers and convert them into goods of different sort. For the j^{th} producer, a vector $\boldsymbol{y}_j = [y_{j1}, y_{j2}, \ldots, y_{jK}]^T$, $y_{jk} \in \mathbb{N}$, called the production plan vector, where $y_{jk} > 0$ if k^{th} good is an output, and $y_{jk} < 0$ if it is an input, defines the amount of goods that the producer takes as input and produces as output. The maximum output of the k^{th} good obtained from the j^{th} producer will be a function of his input goods and of the available technology to produce the good. This is represented by a production function, $v_{jk} : \mathbb{N}^K \to \mathbb{N}$. Each producer has an initial wealth w_j required to start the production. Given a price vector $\boldsymbol{p} = [p_1, \ldots, p_K]^T$, the objective of the j^{th} producer, $j = 1, \cdots, N_{\mathrm{p}}$, is to choose a production plan vector that maximizes his profit, subject to the constraints on the maximum amount of goods that he can produce. Define $\boldsymbol{y}_j^+ = \{y_{jk} \in \boldsymbol{y}_j | y_{jk} > 0\}$, $\boldsymbol{y}_j^- = \{y_{jk} \in \boldsymbol{y}_j | y_{jk} < 0\}$ and $\boldsymbol{p}_j^- = \{p_k \in \boldsymbol{p} \mid y_{jk} < 0\}$. The optimal production plan vector of each producer is obtained by solving the following

constrained optimization problem:

$$\begin{aligned} \boldsymbol{y}_j^* &= \arg\max_{\boldsymbol{y}_j \in \mathbb{N}^K} \boldsymbol{p}^T \boldsymbol{y}_j, \\ &\text{subject to } y_{jk}^+ \leq v_{jk}(\boldsymbol{y}_j^-), \forall k, \text{ and } (\boldsymbol{p}_j^-)^T \boldsymbol{y}_j^- \leq w_j, \quad j = 1, \cdots, N_{\mathrm{p}}. \end{aligned} \tag{5.6}$$

Walras [95] defined a notion of an equilibrium in such consumer-producer economic markets, called the Walrasian equilibrium, which is most commonly used by economists today.

Definition 1. ***Walrasian Equilibrium :*** *The tuples* $\left(\{\boldsymbol{x}_i^*\}_{i=1}^{N_{\mathrm{c}}}, \{\boldsymbol{y}_j^*\}_{j=1}^{N_{\mathrm{p}}}, \boldsymbol{p}^*\right)$ *of the demand vector, the production plan vector and the price vector in an economy form a Walrasian equilibrium, if and only if*

1. $\boldsymbol{x}_i$ *is a solution to the constrained optimization problem given in Eq. (5.5) at the price* $\boldsymbol{p}$, $\forall\ i$
2. $\boldsymbol{y}_j$ *is a solution to the constrained optimization problem given in Eq. (5.6) at the price* $\boldsymbol{p}$, $\forall\ j$, *and*
3. *the market is clear at the price* $\boldsymbol{p}$, *i.e.,* $\sum_{i=1}^{N_{\mathrm{c}}} x_{ik} = \sum_{j=1}^{N_{\mathrm{p}}} y_{jk}, \forall\ k$.

Under some mild assumptions on the continuity and the monotonicity of the utility and the production functions, it was shown that the Walrasian equilibrium exists for all economies [96] using a fixed point argument. Under a much stronger assumption called the *gross substitutability*[10] condition, the equilibrium is also proved to be unique. The key result of the price theory is that the Walrasian equilibria, although defined as a solution to utility maximization and the profit maximization problems

[10]If there is an increase in the price of one good, then the net demand for other goods does not decrease

of individual agents, will produce *Pareto-optimal*[11] allocations. This result is stated as the following two fundamental theorems.

Theorem 1. ***First Fundamental Welfare Theorem*** - *If the triplet* $(\boldsymbol{p}, \boldsymbol{x}_i, \boldsymbol{y}_j)$*, for* $i = 1, 2, \ldots, N_{\mathrm{c}}$ *and* $j = 1, 2, \ldots, N_{\mathrm{p}}$ *is a Walrasian equilibrium, then the allocations* $\boldsymbol{x}_i$ *and* $\boldsymbol{y}_j$ *for* $i = 1, \ldots, N_{\mathrm{c}}$ *and* $j = 1, \ldots, N_{\mathrm{p}}$ *are Pareto-optimal.*

Theorem 2. ***Second Fundamental Welfare Theorem*** - *In a convex economy*[12]*, if* $\boldsymbol{x}_i$ *and* $\boldsymbol{y}_j$ *for* $i = 1, \ldots, N_{\mathrm{c}}$ *and* $j = 1, \ldots, N_{\mathrm{p}}$ *represent any set of Pareto-optimal allocations, then there exists a price vector* $\boldsymbol{p} \in \mathbb{R}^K$ *such that the tuples* $\left(\{\boldsymbol{x}_i^*\}_{i=1}^{N_{\mathrm{c}}}, \{\boldsymbol{y}_j^*\}_{j=1}^{N_{\mathrm{p}}}, \boldsymbol{p}^*\right)$ *form a Walrasian equilibrium for a suitable choice of initial endowments.*

The market equilibrium problem is to compute a price vector, the corresponding demand vectors, and the production plan vectors for all the agents in the economy such that they form a Walrasian equilibrium. This problem is of considerable interest in Economics, and several works have investigated this problem [97]- [98]. However, finding computationally efficient polynomial time algorithms to compute the equilibrium prices and allocations for a general economic model is still a major research area. Over the last few years, there has been a huge effort in the theoretical computer science community to develop efficient algorithms for computing the equilibria [99], [100], [101], [102], [103]. While few groups have been working on developing polynomial time algorithms for specific markets, the other groups focussed on developing algorithms for computing the approximate equilibrium. One such notion of an approximate equilibrium is ϵ-approximate equilibrium. The tuples $\left(\{\boldsymbol{x}_i^*\}_{i=1}^{N_{\mathrm{c}}}, \{\boldsymbol{y}_j^*\}_{j=1}^{N_{\mathrm{p}}}, \boldsymbol{p}^*\right)$ form an ϵ-approximate Walrasian equilibrium if, for $0 < \epsilon < 1$,

[11] An allocation of goods to agents is defined to be Pareto-optimal if no other allocation of the same goods would be preferred by every agent

[12] The utility and production functions of all the consumers and producers are convex functions

the optimal solutions $\{\boldsymbol{x}_i^*\}_{i=1}^{N_c}$ and $\{\boldsymbol{y}_j^*\}_{j=1}^{N_p}$ are such that $\sum_{i=1}^{N_c} x_{ik}^* = (1-\epsilon)\sum_{j=1}^{N_p} y_{jk}^* \forall$ k at price $\boldsymbol{p}^*$, i.e., the market clearing condition is approximately satisfied. In the next subsection, we describe auctions and propose an auction mechanism that can be used to compute an ϵ-approximate equilibrium of the market model described in this subsection.

5.3.2 Auctions and Price discovery

Auction algorithms first originated as methods for finding solutions to an assignment problem where several agents were competing for various resources [104]. Since then, auctions have been used for solving a wide variety of problems in the areas of computer science [105], Economics [106] and finance. Auction-based algorithms are used for two important reasons. First, they are intuitive and easy to implement. Second, they provide a general theoretic framework for understanding the interaction between self-interested agents and provide computationally efficient methods for solving the allocation problems among these agents, with the objective of achieving Pareto-optimal outcomes.

There are many types of auctioning mechanisms, each with its own unique characteristics and applications [Chap-10, [107]]. Four primary types of auctions that are widely used are the ascending bid auction (English auction), the descending bid auction (Dutch auction), the first price sealed-bid auction and the second price sealed-bid auction (Vickrey auction). Based on these primary types, several secondary type auctions have been derived by making minor modifications. For a detailed descriptions of auctions, interested readers can refer [Chap-10, [107]]. Our auction algorithm is

a combination of two auction mechanisms: (i) the double auction [108] and (ii) the combinatorial auction [109].

In a traditional auction, an auctioneer is regarded to be either on the sellers' side or the buyers' side. When an auctioneer is on the sellers' side, his main objective is to maximize the sellers' profits while minimizing their cost, whereas if the auctioneer is on the buyers' side, his objective is to maximize buyers' utility and minimize their purchase cost. Both of these scenarios are considered to be one-sided, and hence a third neutral auctioneer scenario is introduced, where the objective is to strike a balance between the two prior cases, with the main goal of maximizing global welfare. These types of auctions are called double auctions (DA). It was shown that DAs are much more efficient than several one-sided auctions [110]. Combinatorial auctions (CA) are a different class of auction mechanisms, where bidders can place bids on the combinations (or bundles) of goods, instead of being limited to bidding on a single item, as happens in most conventional auctions. This ability to bid on several combinations of goods allows the agents to more accurately express their preferences. Combinatorial double auctions (CDA) [110]- [111], which are a combination of DAs and CAs, are most frequently encountered in market-based economies (for e.g. stock exchange markets), and they represent the advantages of combinatorial auctions by allowing bids to be placed on several combinations of goods, and also the double auctions by considering the requirements of both buyers and sellers in the market. The use of auction-based algorithms for computing the market equilibrium was first proposed in [106] and they have been used extensively since then.

We propose an iterative CDA algorithm that can be used for finding the $\epsilon-$approximate Walrasian equilibrium for the market scenario described in the previous section. We restrict ourselves to the class of linear models, where the utility and the production

functions are linear in the demand vector and the production plan vector, respectively, i.e.,

$$u_i(\boldsymbol{x}_i) = \sum_{k=1}^{K} u_{ik} x_{ik}, \ \forall i, \tag{5.7}$$

$$v_{jk}(\boldsymbol{y}_i) = \sum_{y_j \in \boldsymbol{y}^-} v_{jk} y_{jk}, \ \forall j. \tag{5.8}$$

The run of the algorithm is partitioned into several iterations. Each iteration is further partitioned into three steps. We start with an arbitrary, but fixed order, for each consumer and producer in the economy. Let $\boldsymbol{p}^n = (p_1^n, \ldots, p_K^n)$ denote the price vector at the n^{th} iteration of the algorithm. In the first step of each iteration, the optimal demand vector of every consumer is evaluated. In order to compute the optimal demand vector of the i^{th} consumer, we will use a branch-and-bound [112] technique to solve Eq. (5.5) at price $\boldsymbol{p}^n$. In the second step of each iteration, the optimal production plan vector of every producer is evaluated. The optimal production plan vector is obtained by solving Eq. (5.6) using a branch-and-bound technique. In the third step, the auctioneer will compute the total demand and supply for each of the K goods, and will adjust the price based on the demand and the supply values. The price of the k^{th} good is increased if the demand of the good exceeds the supply, and the price is decreased if the supply exceeds the demand. The increase or the decrease in the price is proportional to the value of the excess demand or the excess supply, respectively, computed at the current price. As the price vector changes, the algorithm recomputes the demand and the production plan vector for each consumer and producer. The algorithm will terminate when an approximate equilibrium is

achieved or when the number of iterations exceeds a predetermined threshold. The detailed auction mechanism is shown in Algorithm 4.

Algorithm 4 Auction Algorithm to find Walrasian Equilibrium of a Double-Sided Market

Initialize $p_k^1 = 0, \forall k$, and $n = 1$
while $n < N_{\text{th}}$ or ϵ-approximate equilibrium not achieved **do**
  Find $\boldsymbol{x}_i^n$ and $\boldsymbol{y}_j^n$ by solving Eqs. (5.5) and (5.6), respectively for all
  if $(\sum_{i=1}^{N_{\text{c}}} x_{ik}^n > \sum_{j=1}^{N_{\text{p}}} y_{jk}^n)$ **then**
    set $p_k^{n+1} = p_k^n(1 + \delta p_k^n|\sum_{i=1}^{N_{\text{c}}} x_{ik}^n - \sum_{j=1}^{N_{\text{p}}} y_{jk}^n|)$
  else if $(\sum_{i=1}^{N_{\text{c}}} x_{ik}^n < \sum_{j=1}^{N_{\text{p}}} y_{jk}^n)$ **then**
    set $p_k^{n+1} = p_k^n/(1 + \delta p_k^n|\sum_{i=1}^{N_{\text{c}}} x_{ik}^n - \sum_{j=1}^{N_{\text{p}}} y_{jk}^n|)$
  else
    set $p_k^{n+1} = p_k^n$
  end if
end while

In this manner, an $\epsilon-$approximate equilibrium of the market is reached via the interactions between the producers and consumers using an auction mechanism. We make several comments about the auction mechanism here. First, we assume that the processing center has sufficient computational and communication resources to execute the mechanism and maintain the equilibrium in real-time. Second, there are two methods to implement the auction mechanism across the sensor network: distributed implementation and a centralized implementation. In a distributed implementation, each agent controls the physical entity it represents, and is capable of computation. In our market, a distributed implementation means that the consumers and the producers evaluate their respective optimal demand and production plan vectors at a given price, and submit the demand and production plan vectors in the form of bids to an auctioneer. The auctioneer will then adjust the price based on the demand and the supply. In a centralized implementation, all the processing is done at a central unit. In our market, a centralized implementation means that the producers and consumers

will inform their utility and production functions to the processing center, which then evaluates their optimal production and demand vectors. The processing center can itself act as an auctioneer, and adjust the prices for the subsequent iterations. The distributed implementation is communication intensive as the producers, consumers, and the auctioneer should exchange bids and prices at each iteration, whereas the centralized implementation is computation intensive as the processing center has to evaluate the optimal demand and production plan vectors for all the agents in the market. In this work, we adopt a centralized implementation for two important reasons which we describe in the next Section. Third, we assume that all the agents in the market are price takers. This assumption ensures that the equilibrium point is a Pareto-optimal allocation.

5.4 Sensor Management for MTT

In this section, we address sensor management in multi-modal networks for the MTT problem. Although the description is specific to target tracking, the proposed framework can be easily extended for sensor management in other applications employing multi-modal sensors. To keep the description simple, we address SS, RA and DF using a simple particle filter. Our approach to sensor management is based on modeling the interaction between the SNs and the SM as an interaction between the agents in an economic market. Each agent acts as a self interested unit, with the goal of maximizing its utility. As the utilities of the agents keep changing, they re-evaluate their preferences and reach an equilibrium. At the equilibrium, none of the agents will deviate from their respective preferences due to the Pareto-optimality at the equilibrium.

We assume that all the data processing is done by the SM. We model the SM as a consumer in the market that purchases measurements from each of the sensors and sells the resources it has to the sensors. In this chapter, we consider a sensor network that comprises of a multistatic radar, an infrared camera and a human scout. The state model is same as the one employed in Chapter 4. The measurement models of the infrared camera and the human scout are also same as the ones used in Chapter 4. We modify the measurement model of the radar system slightly, to include three transmit antennas. We provide the measurement models for the sensors below.

1. Radar sensor:

$$\boldsymbol{y}_{p,t} = \sum_{n_t=1}^{N_t} \sum_{q=1}^{3} \beta_{p,q,n_t} \sqrt{\gamma_q \zeta_{p,q,n_t}} \Big(\boldsymbol{\Upsilon}(p,q,n_t) \otimes \boldsymbol{\Gamma}(p,q,n_t) \Big) \boldsymbol{s} + \boldsymbol{w}_p, p = 1,2,3 \; q = 1,2,3. \tag{5.9}$$

2. Infrared camera:

$$p(\boldsymbol{y}_{4,t} \mid \boldsymbol{\theta}_t) = \mathcal{N}(\boldsymbol{y}_{4,t}; \gamma_4 \boldsymbol{I}_0, \boldsymbol{\Sigma}_{\text{ir},1} \otimes \boldsymbol{\Sigma}_{\text{ir},2}). \tag{5.10}$$

3. Human scout:

$$p(\boldsymbol{y}_{5,t}|\boldsymbol{\theta}_t) = \begin{cases} p_{\text{c}} \sum_{k=1}^{N_{\max}} g(k) \frac{k!}{y_{5,t,1}! \cdots y_{5,t,M}!} q_{1,t}^{y_{5,t,1}} \cdots q_{M,t}^{y_{5,t,M}} & \text{if at least one target is incorrectly identified,} \\ (1-p_{\text{c}}) + p_{\text{c}} [g(N_{\text{hs,t}}) \frac{N_{\text{hs,t}}!}{y_{5,t,1}! \cdots y_{5,t,M}!} q_{1,t}^{y_{5,t,1}} \cdots q_{M,t}^{y_{5,t,M}}] & \text{otherwise.} \end{cases} \tag{5.11}$$

In the above, the constants $\gamma_1, \ldots, \gamma_5$ in the measurement models of the radar, the infrared camera and the human scout, respectively, determine the quality of the measurements that these sensors obtain. These constants can be thought of as resources that the SM provides the SNs. However, in practice, not all SNs can obtain good data at same time, since the SM has only a finite amount of resources. For example,

if the SM spends some amount of resource in training the scout, it might not have enough resources left to obtain a good camera, or to transmit the waveforms with higher energy. Hence, we enforce a constraint of the form $\sum_{p=1}^{5} \gamma_p \leq \gamma$, where γ is the constraint on the available resource. We call these constants $\gamma_1, \ldots, \gamma_5$ as the power allocated to the respective sensors and γ as the total available power. Let $\boldsymbol{x}_t = [x_{1,t}, \ldots, x_{6,t}]^T$ denote the demand vector of the SM at time t. Note that since there is only one consumer in the market, we dropped the subscript for consumer index. The demand vector $\boldsymbol{x}_t$ comprises two parts: the number of measurements that the SM seeks from each SN and the power that the SM distributes to the SNs. Let the number of measurements that the SM seeks from the p^{th} sensor at time t be denoted $x_{p,t}, p = 1, \ldots, 5$. In general, $x_{p,t} = \{0, 1\}, p = 1, \ldots, 5$, which means that the p^{th} sensor is either inactive or it collects a single measurement. However, using diversity techniques, such as time, space and frequency, independent realizations of the measurement vectors can be obtained. Therefore, we consider that $x_{p,t} \in \mathbb{N}$. Denote the total power that the SM distributes be denoted using $x_{6,t}$. Note that $x_{p,t} \geq 0$, for $p = 1, \ldots, 5$, and $x_{6,t} < 0$. The number of measurements that the SM seeks correspond to the quantity of goods that the SM is willing to purchase, and the power allocated corresponds to the quantity of the good that the SM is willing sell. The optimal demand vector of the SM can be obtained by solving

$$\begin{aligned} &\boldsymbol{x}_t^* = \arg \max_{\boldsymbol{x}_t \in B_t(\boldsymbol{p}_t, \boldsymbol{e}_t)} u_t(\boldsymbol{x}_t), \\ &\text{where } B_t(\boldsymbol{p}_t, \boldsymbol{e}_t) = \{\boldsymbol{x}_t \in \mathbb{N}^K : \boldsymbol{p}_t^T \boldsymbol{x}_t \leq \boldsymbol{p}_t^T \boldsymbol{e}_t\}, \end{aligned} \tag{5.12}$$

where $e_{p,t} = 0, p = 1, \ldots, 5$, and $e_{6,t} = \gamma$ is the total available power. We consider an information theoretic utility function to characterize the preference of the SM to the various choices of the measurements that it can obtain from the sensors. Since

the overall goal of the SM is to estimate the unknown target state, we chose a utility function that reduces the uncertainty about the unknown state vector $\boldsymbol{\theta}_t$.

$$u_t(\boldsymbol{x}_t) = -\sum_{p=1}^{5} \frac{1}{d_p} x_{p,t} \mathrm{H}(\boldsymbol{\theta}_t | \boldsymbol{y}_{p,1:t-1}), \tag{5.13}$$

where d_p is the dimension of the subspace that the p^{th} sensor observes, and $H(z)$ is the entropy [113] of the random variable z defined as $H(z) = -\sum_{z \in \mathcal{Z}} p(z) \log(p(z))$. For the utility function defined in Eq. (5.13), the posterior distribution, $p(\boldsymbol{\theta}_t | \boldsymbol{y}_{p,1:t-1})$ can be approximated using a set of N_p particles $\left\{\boldsymbol{\theta}_t^{(k)}\right\}_{k=1}^{N_\mathrm{p}}$ and associated weights $\left\{w_{p,t-1}^{(k)}\right\}_{k=1}^{N_\mathrm{p}}$. This method of approximating the posterior distribution is employed in particle filtering, and we will describe the particle filtering in detail in the next subsection. With this approximation, the utility function can be simplified as

$$u_t(\boldsymbol{x}_t) = \sum_{p=1}^{5} \frac{1}{d_p} x_{p,t} \sum_{k=1}^{N_\mathrm{p}} w_{p,t-1}^{(k)} \log(w_{p,t-1}^{(k)}). \tag{5.14}$$

We model the SNs as producers in the market that obtain measurements by using the resources that the SM provides them. In the multiple target tracking example, there are five producers: the three radar antennas, the infrared camera and the human scout. Let $v_{j,t}(y_{j,t}^-)$ denote the production function of the j^{th} producer. The production function defines the number of measurements that the producer can obtain, as a function of the resources allocated to it. We consider the following linear production function for each producer at time t

$$v_{j,t}(y_{j,t}^-) = c_j y_{j,t}^-, \tag{5.15}$$

where $v_{j,t}(y_{j,t}^-)$ represents the maximum number of measurements that j^{th} sensor can obtain at time t, $y_{j,t}^-$ represents the resource allocated to the j^{th} sensor at time t, and c_j is a known constant. In this work, we chose $c_j = 1, j = 1, \ldots, 5, \forall\ t \in \mathbb{N}$. The optimal production plan vector of each producer is obtained by solving

$$\begin{aligned} \boldsymbol{y}_{j,t}^* &= \arg \max_{\boldsymbol{y}_{j,t} \in \mathbb{N}^K} \boldsymbol{p}_t^T \boldsymbol{y}_{j,t}, \\ &\text{subject to } y_{j,t}^+ \leq v_{j,t}(\boldsymbol{y}_{j,t}^-) \text{ and } (\boldsymbol{p}_{j,t}^-)^T \boldsymbol{y}_{j,t}^- \leq w_j \quad j = 1, \ldots, 5. \end{aligned} \tag{5.16}$$

In this manner, we create an artificial market for the sensor information and the resources. The SNs and the SM act as agents in the market, and these agents interact in the market to reach an equilibrium. We use the auction mechanism described in Table 4 to obtain an ϵ-approximate equilibrium point $\left(\{\boldsymbol{x}^*\}, \{y_j^*\}_{j=1}^5, \boldsymbol{p}^*\right)$ for this market. In the equilibrium solution thus obtained, $\boldsymbol{x}^*$, which corresponds to the number of the measurements obtained from each sensor, is the solution to the SS problem; $(y_j^*)^+$, which corresponds to the power allocated to each sensor, is the solution to the RA problem; and $\boldsymbol{p}^*$, which corresponds to the price of the measurements obtained by each sensor, is the solution to the DF problem. In our work, we use centralized processing for two reasons. First, the SNs in the network collect the data, but they do not have any processing capabilities. Hence, they cannot compute their optimal production vectors. Second, bandwidth is scarce in any sensor network and exchanging the bids and prices will increase the overhead of the system in terms of communication bandwidth. Therefore, the MTT is more suited for a centralized implementation than a distributed implementation. Since the production functions do not change with time, the SNs can communicate their production functions to the SM in an offline fashion at the beginning of the tracking process. Hence, there is no information exchange during the sensor management.

Tracking Algorithm

We use a standard particle filter [18] to compute an estimate of the state vector. First, we compute the global posterior distribution as a linear sum of the local posterior distributions, given the measurements collected by the individual sensors using linear opinion pool [75].

$$p(\boldsymbol{\theta}_t \mid \boldsymbol{Y}_{1:t}) = \sum_{p=1}^{P} \mu_p p(\boldsymbol{\theta}_t \mid \boldsymbol{y}_{p,1:t}). \tag{5.17}$$

Here, $P = 5$. We used a simple particle filter where the $\left\{\boldsymbol{\theta}_t^{(k)}\right\}_{k=1}^{N_\mathrm{p}}$ are drawn according to the state-transition distribution and the weights are derived using the principle of importance sampling. The minimum mean-squared error (MMSE) estimate of the state vector is then obtained as the mean of the global posterior distribution as

$$\hat{\boldsymbol{\theta}}_t = \sum_{p=1}^{P} \mu_p \sum_{k=1}^{N_\mathrm{p}} w_{p,t}^{(k)} \boldsymbol{\theta}_t^{(k)}. \tag{5.18}$$

5.5 Numerical Results

In this section, we use numerical examples to demonstrate the performance improvement obtained due to sensor management using the proposed price theory framework. In order to quantify the performance of the multiple target tracking system, we define four performance metrics. We describe the simulation setup first and then discuss the examples.

Target parameters: We consider surveillance of a region for a period of 20 tracking intervals. The duration of each tracking interval was 0.1 s ($\Delta t = 0.1s$). We consider tracking under two scenarios. In the first scenario, during the first 0.8 s, which

Table 5.1: Table showing the initial positions and velocities of all the target categories

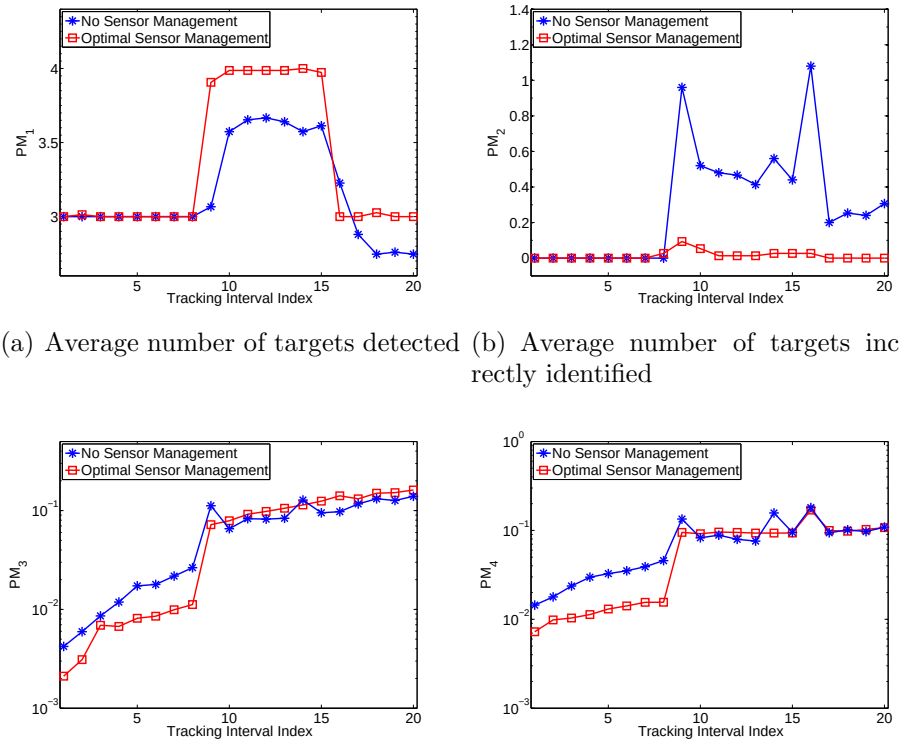
Target Category	Initial Position	Initial Velocity
1	(5,5)	(10,5)
2	(15,10)	(5,0)
3	(10,10)	(5,10)
4	(40,40)	(-10,-10)
5	(0,40)	(5,-10)

corresponds to 8 intervals, there were 3 targets in the scene, during the next 0.8 s, i.e., between the 9^{th} and the 15^{th} interval there were 4 targets, and thereafter there were 3 targets again. The number of target categories was chosen to be 5 ($M = 5$), and the initial positions and initial velocities of the targets belonging to various categories were chosen as shown in the Table 5.1. During the entire tracking duration, the target categories were chosen as

$$\boldsymbol{\alpha}_t = \begin{cases} \{1, 2, 5\} & t = 1, 2, \ldots, 8 \\ \{1, 2, 5, 3\} & t = 9, 10, \ldots, 15 \\ \{1, 2, 3\} & t = 16, \ldots, 20 \end{cases}$$

In the second scenario, there were three targets during the entire duration and the categories of these targets were chosen to be $\boldsymbol{\alpha}_t = \{1, 2, 4\}$ for $t = 1, \ldots, 20$. The initial positions and velocities of these targets were again chosen according to Table 5.1. The probabilities of the birth and the death of the targets were chosen to be 0.01, i.e., $p_{\mathrm{d}} = p_{\mathrm{b}} = 0.01$, respectively.

Signal and Sensor Parameters: We transmit OFDM [114] waveforms with eight ($G = 8$) subcarriers loaded with same symbol in all the subcarriers from the radar antennas. The total bandwidth was 100Mhz ($B = 100$) and the carrier frequency, f_{c}, of the transmitted waveforms was 1Ghz. We used four ($L = 4$) pulses in each



(a) Average number of targets detected (b) Average number of targets incorrectly identified

(c) Average RMSE in range (d) Average RMSE in velocity

Figure 5.1: Performance comparison of the two approaches for the first scenario.

tracking interval. The radar antennas were located at $(x_1, y_1) = (0, 0), (x_2, y_2) = (20, 0), (x_3, y_3) = (40, 0)$, respectively, and the variance of the measurement noise at each antenna was $\sigma^2_{\mathrm{w},p} = 1 \times 10^{-3}, p = 1, 2, 3$. The covariance matrices of the measurement noise at the infrared camera along the row and columns, respectively, were chosen to be $\boldsymbol{\Sigma}_{\mathrm{ir},1} = \sigma^2_{\mathrm{ir}} \boldsymbol{I}_R, \boldsymbol{\Sigma}_{\mathrm{ir},2} = \sigma^2_{\mathrm{ir}} \boldsymbol{I}_C$, with $\sigma^2_{\mathrm{ir}} = 1 \times 10^{-2}$. The constants b and q_c for the human scout were chosen to be 0.2 and 0.05, respectively. We assumed that all the targets were observable by all the sensors.

We evaluated the performance of the system using four metrics that we used in Chapter 4: the average number of targets detected in the scene, the average number of targets identified incorrectly, the root mean-squared error (RMSE) in the position

of correctly identified targets, and the root mean-squared error in the velocity of correctly identified targets.

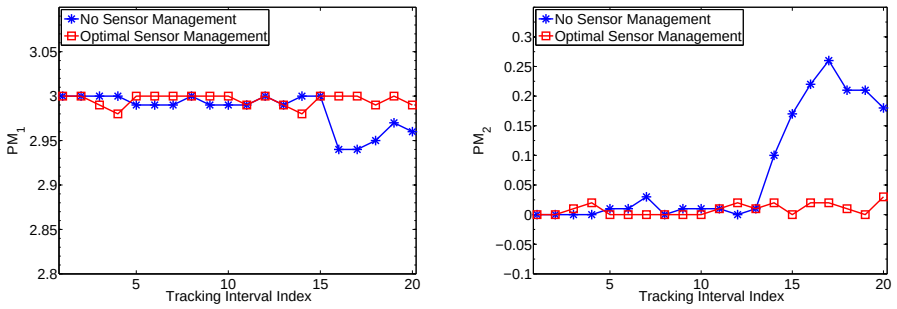
In Fig. 5.1, we plot these metrics for two methods. In the first method, which we refer as standard approach, there was no sensor management. We used all sensors at all times, collected one measurement from each sensor, and equally distributed the power among them. The particle weights corresponding to each sensor were updated following standard weight update equation, and the global posterior density was computed following Eq. (4.8) by giving equal weights to the sensors, i.e., $\mu_p = 1/P, p = 1, \ldots, P$. In the second method, which we refer as price-theory approach, we used the proposed price theory framework to select the number of measurements that the each sensor should obtain, and to allocate power to the selected sensors using the demand and the production plan vectors of the SM and the SNs, respectively. Further, we computed the price of the measurements collected by the individual sensors, and used the price of data as the weight given to the corresponding sensor to evaluate the global posterior density (see Eq. (5.17)). During the interaction between the SNs and the SM, the SM will request for a higher number of measurements from the SN which improves the utility function of the SM. In order to meet the demand, this SN will increase the price of data it collects. As a result, the SNs which improve the utility function will give a higher price to the data they collect, and therefore, the price of the data can act as a measure of importance of the data.

In order to find the demand vector, the production plan vector, and the price, i.e., $\{\boldsymbol{x}, \{\boldsymbol{y}_j\}_{j=1}^5, \boldsymbol{p}\}$, we computed the ϵ-approximate Walrasian equilibrium for market model using Algorithm 4. We used a branch-and-bound technique [112] to solve the constrained optimization problems given by Eqs. (5.12) and (5.16). For the auction algorithm, we chose $\epsilon = 3$ and $N_{\text{th}} = 1000$. To find the estimates of the state vector,

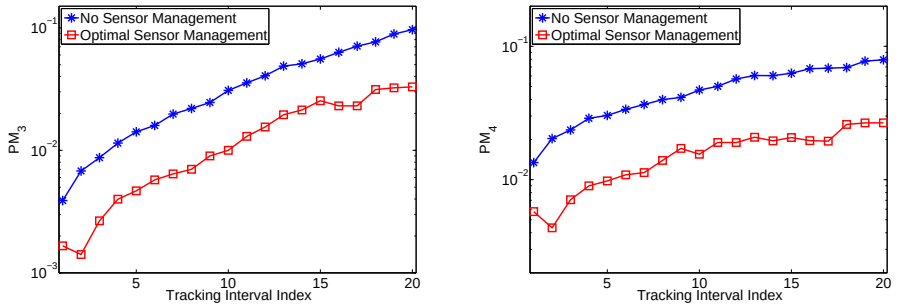
we used a particle filter with $N_{\mathrm{p}} = 1500$ particles. The simulations were averaged over $N_{\mathrm{mc}} = 100$ Monte-carlo iterations.

We can see from Fig. 5.1(a) that the system was able to accurately estimate the number of targets when sensor management was used. On the other hand, not using sensor management resulted in an incorrect estimation of the number of targets. It can also been seen that employing sensor management resulted in higher correct identifications of the target categories compared to the case when sensor management was not used. However, from Figs. 5.1(c) and 5.1(d), it can be seen that the average RMSE in the range and the velocity, per target, for the correctly identified targets, using sensor management, increased after a few iterations and was comparable to the average RMSE obtained without using sensor management. This is because when the number of targets changes in a particular tracking interval, there are a very few particles that correspond to the changed target number. A price theory approach to sensor management gives higher weights to these particles and computes the state estimates based on these fewer particles. As a result, the RMSE in the range and the velocity estimates increases. When sensor management is not used, the actual target number is not estimated correctly. As a result, there are several particles, that do not correspond to the actual target state, that get a higher weight which results in a lower RMSE in this case. Hence, the RMSE using the price theory approach for sensor management increases at the intervals where the number of the targets change.

In Fig. 5.2, we plot the performance metrics for the second scenario. For this case, it can be seen that using sensor management resulted in an accurate estimate of the number of targets, higher correct identifications, and a lower average RMSE in range and velocity estimates. As the number of targets remained same throughout



(a) Average number of targets detected (b) Average number of targets incorrectly identified

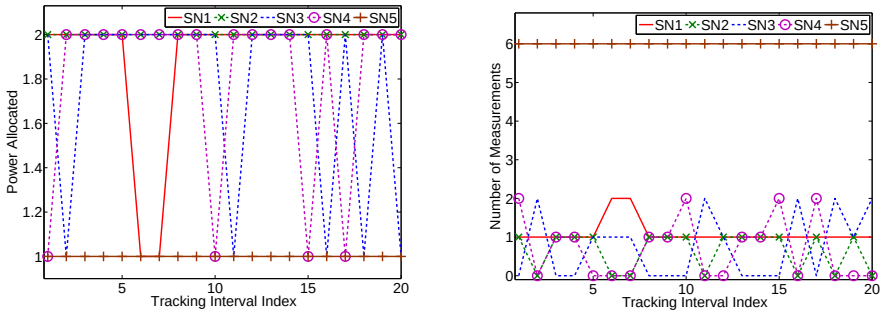


(c) Average RMSE in range (d) Average RMSE in velocity

Figure 5.2: Performance comparison of the two approaches for the second scenario.

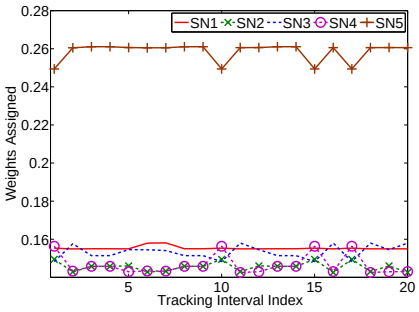
the tracking period, there was no sudden increase in RMSE of the range and velocity estimates unlike the first scenario.

In Figs 5.3 and 5.4, we plot the output of the sensor selection, resource allocation and data fusion for the two scenarios that we considered. The SM seeks more measurements from the human scout compared to the other sensors. This is expected since the scout is trained in accurately counting the number of targets and their categories compared to other sensors. The power allocated to the other sensors and the weights assigned to them change with the utility function.



(a) Average power allocated to each sensor

(b) Average number of measurements obtained from each sensor

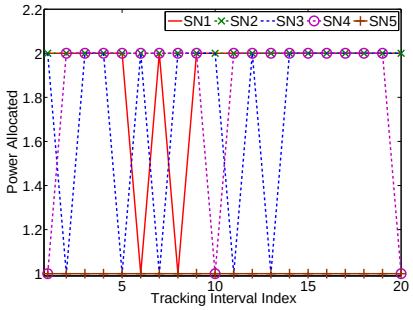


(c) Average weight given to each sensor

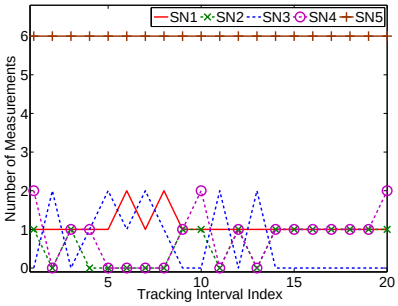
Figure 5.3: Solutions to the SS, RA, and DF problems for the first scenario.

5.6 Summary

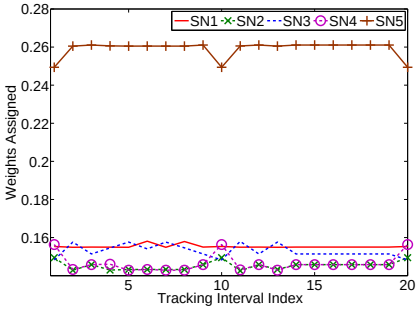
We considered the problems of sensor selection (SS), resource allocation (RA) and data fusion (DF) that comprise the sensor management in multi-modal networks, and we developed a framework, based on economic price theory, to jointly solve SS, RA, and DF. We illustrated this framework using a scenario where the task is to track an unknown and time-varying number of targets, using the measurements obtained by three different kinds of sensors: a multistatic radar, an infrared camera and a human scout. Numerical examples showed that the proposed framework for sensor management was effective in estimating the number of targets, accurately identifying the target categories, and producing lower root mean-squared error in the range



(a) Average power allocated to each sensor



(b) Average number of measurements obtained from each sensor



(c) Average weight given to each sensor

Figure 5.4: Solutions to the SS, RA, and DF problems for the second scenario.

and velocity estimates of the targets when compared to the root mean-squared error obtained when sensor management was not employed.

Chapter 6

Concurrent Particle Filtering for MTT in the Presence of Association Errors[13]

In this chapter, we develop a filtering method that addresses the issue of data association ambiguity that arises in multiple-target tracking.

6.1 Introduction

MTT becomes challenging when the measurements at the receiver(s) comprises measurements due to moving targets and those due to clutter, which are generally considered false alarms. Some times, due to the low target detection probability, some targets do not produce any measurements during a scan. As a result, before the state estimation, we need to identify which measurements are target-originated and which

[13]Based on P. Chavali and A. Nehorai, "Concurrent particle filtering and data association using game theory for tracking multiple maneuvering targets", *in IEEE Trans. on Signal Processing*, vol. 61, pp. 4934-4948, Oct. 2013.

measurements are clutter-originated. This problem of assigning a measurement to each target is called the data association. When the number of targets and clutter rate increase, the data association becomes exponentially more complex [1].

When the targets themselves are moving according to various kinematic models (which we will call as modes from now on), in addition to associating measurements to targets and estimating the target state, we also needs to estimate the mode of each target at each point in time. With time, the problem of keeping track of target kinematics also grows exponentially hard [1]. Approximate methods such as the interacting multi-model (IMM) filters [115],where a bank of filters with each filter tuned to a specific mode, are often used to solve this problem. Monte-Carlo methods for target mode tracking [116], [115] are also available to overcome the exponential complexity, and it has been shown that the IMM particle filters offer a better performance compared to the traditional IMM filters [117].

6.1.1 Review of Related Work

All the existing multiple target tracking algorithms that consider the problem data association can be divided into two main classes [1]. Unique-neighbor data association methods that associate each measurement to one of the existing targets (tracks), and all-neighbors data association methods that use all the measurements for updating the track of each target. Nearest neighbor (NN) [1] is one of the simplest methods of the data association. In NN, at each time step, a single nearest measurement is associated to each target by assuming that all the other measurements are generated from clutter. The NN algorithm performs a local optimization in that it operates target by target. This can lead to several targets being associated with

the same measurement, which might lead to poor tracking performance. The global version of the nearest neighbor (GNN), searches for the best global association, considering all targets and measurements simultaneously. Multiple Hypothesis testing (MHT) [118], another unique-neighbor association method, carries forward all the association hypothesis to the next time step and aggregates them over the time. A best possible hypothesis for a previous time is evaluated in retrospect. The probabilistic data association (PDA) framework [119] is an all-neighbor data association method that associates a linear combination of measurements to each target and the joint probabilistic data assocoation (JPDA) [120] finds the association probabilities by considering all the targets and the measurements simultaneously. JPDA requires an exhaustive enumeration of all possible associations at the current time step, which can lead to exponential complexity. Hence, gating is used to reduce the number of associations to a feasible level. Probabilistic MHT (PMHT) [121] is another filtering technique where the association probabilities of each target are assumed to be independent, and are evaluated independently, thereby eliminating the need for gating. Unfortunately, none of the above algorithms cope with nonlinear and non-Gaussian models.

Recently, there has been a surge of work using the SMC based approaches in multiple-target tracking. SMC methods belong to the class of unique-neighbor data association methods, and they can be applied in non-linear and non-Gaussian scenarios. There are several variations of the SMC methods that are used. In [122], the authors develop a hybrid bootstrap filter in which the resampling step is replaced by a step that fits a finite mixture distribution to the posterior samples, and the samples are drawn from the mixture density; in [123], data-association is considered as a missing data problem, and Gibbs sampling is used in order to obtain samples of the unknown

association variable; in [124], and [82], Monte Carlo approaches to JPDA filtering are proposed; in [125], a Rao-blackwellized Monte Carlo data association is proposed for the case of conditional linear models; and in [82], the authors propose two particle filter architectures called sequential sampling particle filter (SSPF) and the independent partition particle filter (IPPF). In order to draw samples of the target state from a proposal distribution, all the above methods either employ sequential sampling, in which samples are drawn for individual targets sequentially, or invoke an independent target association assumption and draw samples for each target independently (for example, in IPPF). Although the independence assumption addresses the computational limitations of the sequential sampling, it has serious performance degradation, especially in the cases where the probability of detection is low and the clutter rate is high.

In this chapter, we extend the SSPF and the IPPF developed in [82] to the case of maneuvering targets. We combine the Monte-Carlo approaches for data association and multi-mode particle filters to design filters that can track multiple maneuvering targets in the presence of data association ambiguity. We describe the choice of the target and the association state proposals and then derive the weight update equations. We call the resulting filters the IMM-SSPF and the IMM-IPPF, respectively.

Next, we propose a new particle filter, which we refer to as the interacting multi-model concurrent particle filter (IMM-CPF) to track maneuvering targets. IMM-CPF combines the advantages of both IMM-SSPF and IMM-IPPF architectures to obtain better tracking accuracy and lower implementation complexity. The key difference between the proposed filter and the existing SSPF/IPPF architectures is that we develop and employ a game-theoretic formulation to solve the data association in a deterministic fashion. Once the associations are known, each target can be tracked

independently and simultaneously (hence the name concurrent). Since our methods uses a deterministic approach to find the data association, it follows hard-gating. We emphasize here that our approach is different from the *distributed particle filtering* [126], [127], [128], [129] in the sense that the goal of our approach is to track all the targets simultaneously using the measurements from a sensor(s), whereas in a distributed particle filtering the goal is to track a target(s) using measurements from several interacting sensors.

To solve the association ambiguity, we formulate the problem of data association as a game between multiple trackers. We specify the strategy set and the utility functions of each tracker. We then use a regret-based learning algorithm called regret matching to find the equilibrium this game in a distributed manner. The distributed implementation enables all the trackers to simultaneously assign a measurement to an existing target track, without making an assumption of independent association.

6.2 System Model

In this section, we formulate the problem of multiple maneuvering target tracking in the presence of data association ambiguity. We first describe the state model for maneuvering targets, and then describe the measurement model that considers association ambiguities.

6.2.1 State-Space Model

We will provide a brief description of the state-space model for maneuvering targets in this section. We assume that the number of targets is known, which is denoted by K. For the k^{th} target, we denote the target state at time t as $\boldsymbol{\theta}_{k,t} = [x_{k,t}, y_{k,t}, \dot{x}_{k,t}, \dot{y}_{k,t}]^T$, where $x_{k,t}, y_{k,t}$ are the x-position and the y-position at time t, and $\dot{x}_{k,t}, \dot{y}_{k,t}$ are the x-velocity and y-velocity, respectively.

Unlike the state models used in other chapter, we assume that the target kinematics follow multiple modes in this chapter. In this model, each target can switch the kinematic modes between the M possible modes (e.g. stopped, moving with constant velocity, accelerating, or co-ordinated turn). The switch is assumed to happen instantaneously. Let $\mathcal{M} = \{1, 2, \ldots, M\}$ denote the set of all possible modes of the target and $\alpha_{k,t} \in \mathcal{M}$ denote the mode of the k^{th} target at time t. The temporal evolution of the k^{th} target state is then given by the state transition equation

$$\boldsymbol{\theta}_{k,t} = g(\boldsymbol{\theta}_{k,t-1}, \alpha_{k,t}, \boldsymbol{v}_{k,t-1}),\ k = 1, \ldots K,\ t \in \mathbb{N}, \tag{6.1}$$

where $\boldsymbol{v}_{k,t}$ is the process noise modeled as white Gaussian with the covariance matrix $\boldsymbol{\Sigma}_{\mathrm{v},k}$. Note that in general, the process noise can be mode-dependent. For simplicity, we assume that the process noise does not depend on the mode. Further, we assume that the state transition equations corresponding to individual targets are independent of each other.

For each target, we model the mode transition using a stationary Markov chain, where the transition probabilities are given as

$$\Pr(\alpha_{k,t} = m \mid \alpha_{k,t-1} = n) = p_{nm}, \tag{6.2}$$

with $\boldsymbol{P} = [p_{nm}]$. Note that we assumed the mode transition probabilities to be same for all the targets, for simplicity. The mode transition probabilities are known *a priori*. At a given time, we use $\boldsymbol{\pi}_{k,t} = [\pi_{k,t}^1, \ldots, \pi_{k,t}^M]^T$ to represent the probabilities with which the k^{th} target follows each of the M modes, i.e., $\pi_{k,t}^m = \Pr(\alpha_{k,t} = m)$. The initial probability vectors $\boldsymbol{\pi}_{k,0}$ are assumed to be known for all the targets.

6.2.2 Measurement Model

We assume that the measurements are collected by a single sensor located at the origin of the coordinate system. At time t, the sensor receives a measurement vector $\boldsymbol{y}_t = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{N_t}\}$, where N_t is the number of measurements collected at time t. Note that the number of measurements varies with time. The measurement vector $\boldsymbol{y}_t$ comprises two parts: the detection measurements from the targets and the false alarms from the clutter. Let N_t^{d} be the number of detections and N_t^{c} be the number of false alarms. As we do not know the origin of each measurement, an additional variable, called the association variable, is introduced to describe the associations between the targets and the measurements. For each target, we define the association variable $\lambda_{k,t}$ as:

$$\lambda_{k,t} = \begin{cases} n & \text{if the } k^{th} \text{ target generates } n^{th} \text{ measurement} \\ 0 & \text{if the } k^{th} \text{ target is not detected.} \end{cases} \tag{6.3}$$

Note that in our definition of the association variable, we associate each target to a measurement. This definition is different from the commonly used definition, which associates each measurement to a target. However the two approaches are equivalent [82]. From the definition of the association variable, it can be seen that the association

variable for two different targets should be different, since no two targets can be associated to same measurement. We use feasibility of the association vector as a means to characterize this property.

Definition 2. *An association vector* $\boldsymbol{\lambda}$ *is called a feasible association vector if for* $k_1 \neq k_2$*, either* $\lambda_{k_1} \neq \lambda_{k_2}$ *or* $\lambda_{k_1} = \lambda_{k_2} = 0$*.*

If the n^{th} measurement is obtained from the k^{th} target, i.e., $\lambda_{k,t} = n$, we model the measurement as

$$\boldsymbol{y}_{n,t} = f(\boldsymbol{x}_{k,t}) + \boldsymbol{w}_{n,t}, \tag{6.4}$$

where $\boldsymbol{w}_{n,t}$ is the additive measurement noise, that is assumed to be white Gaussian with the covariance matrix $\boldsymbol{\Sigma}_{\mathrm{w}}$. Further, for $n_1 \neq n_2$, the stochastic noise processes $w_{n_1,t}$ and $w_{n_2,t}$ are assumed to be independent of each other. Therefore we have,

$$p(\boldsymbol{y}_{n,t} \mid \lambda_{k,t} = n, \boldsymbol{\theta}_{k,t}) = \mathcal{N}\Big(\boldsymbol{y}_{n,t}; f(\boldsymbol{\theta}_{k,t}), \boldsymbol{\Sigma}_n\Big). \tag{6.5}$$

We model the number of clutter measurements as a Poisson random variable, with parameter γV, where V is the volume of the observation area, and the number of detection measurements as a Binomial random variable. We have

$$\Pr(N_t^{\mathrm{c}} = n_1) = \frac{(\gamma V)^l e^{-\gamma V}}{l!}, \text{ and,} \tag{6.6}$$

$$\Pr(N_t^{\mathrm{d}} = n_2) = \begin{pmatrix} K \\ n_2 \end{pmatrix} P_{\mathrm{d}}^{n_2} (1 - P_{\mathrm{d}})^{K - n_2}, \tag{6.7}$$

where P_{d} is the probability of detection for a single target. The clutter measurements are uniformly distributed over the measurement area, and their likelihood is given by

$$p(\boldsymbol{y}_{n,t} \mid n \notin \boldsymbol{\lambda}_t) = \frac{1}{V}. \tag{6.8}$$

In most practical applications, the association variables are not known, and they need to be estimated along with other unknowns. In order to solve the data association, some assumptions are commonly made [1], [123]:

A1 Each measurement is either a false alarm or is originated from at most one target.

A2 Each target generates at most one measurement at one time. This assumption implies that $\lambda_{k,t}, k = 1, 2, \cdots K$ are dependent random variables.

Some times assumption A2 is replaced by the assumption A3 [130], [131], [132], which is also called as the PMHT assumption, following the name of the filtering technique that employs this assumption.

A3 Each target can generate zero or several measurements at one time. This assumption implies that $\lambda_{k,t}, k = 1, 2, \cdots K$ are independent random variables.

6.3 Data Association using Monte-Carlo Methods

In this section, we describe a solution to MTT using a particle filter. We derive the expressions for the time update and the measurement update in the presence of data association ambiguity for the maneuvering targets. After augmenting the target

state $\boldsymbol{\theta}_t$ with the unknown associations $\boldsymbol{\lambda}_t$, and the unknown mode $\boldsymbol{\alpha}_t$, the time and measurement update equations can be written as

$$\begin{aligned} p(\boldsymbol{\theta}_t, \boldsymbol{\lambda}_t, \boldsymbol{\alpha}_t \mid \boldsymbol{y}_{1:t-1}) &= p(\boldsymbol{\lambda}_t) \times p(\boldsymbol{\alpha}_t \mid \boldsymbol{y}_{1:t-1}) \\ &\times \int p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}, \boldsymbol{\alpha}_t) \sum_{\substack{\boldsymbol{\lambda}_{t-1} \\ \boldsymbol{\alpha}_{t-1}}} p(\boldsymbol{\theta}_{t-1}, \boldsymbol{\lambda}_{t-1}, \boldsymbol{\alpha}_{t-1} \mid \boldsymbol{y}_{1:t-1}) \mathrm{d}\boldsymbol{\theta}_{t-1}, \quad (6.9) \\ p(\boldsymbol{\theta}_t, \boldsymbol{\lambda}_t, \boldsymbol{\alpha}_t | \boldsymbol{y}_{1:t}) &= \frac{1}{z}\, p(\boldsymbol{y}_t \mid \boldsymbol{\theta}_t, \boldsymbol{\lambda}_t) p(\boldsymbol{\theta}_t, \boldsymbol{\lambda}_t, \boldsymbol{\alpha}_t \mid \boldsymbol{y}_{1:t-1}). \quad (6.10) \end{aligned}$$

In the above $p(\boldsymbol{\lambda}_t)$ is the association prior given as

$$\begin{aligned} p(\boldsymbol{\lambda}_t) &= \Pr(N_t^{\mathrm{c}}) \Pr(N_t^{\mathrm{d}}) p(\boldsymbol{\lambda}_t \mid N_t^{\mathrm{d}}, N_t^{\mathrm{c}}), \\ &= \Pr(N_t^{\mathrm{c}}) \Pr(N_t^{\mathrm{d}}) \prod_{k=1}^{K} p(\lambda_{k,t} \mid \lambda_{1:k-1,t}), \quad (6.11) \end{aligned}$$

where

$$p(\lambda_{k,t} = n \mid \lambda_{1:k-1,t}) = \begin{cases} 1 - P_{\mathrm{d}} & \text{if } n = 0 \\ 0 & \text{if } n > 0 \text{ and } n \in \{\lambda_{1,t}, \ldots, \lambda_{k-1,t}\} \\ \frac{P_{\mathrm{d}}}{N_{k,t}^{\mathrm{u}}} & \text{otherwise,} \end{cases} \quad (6.12)$$

and $N_{k,t}^{\mathrm{u}} = N_t - |\{l : \lambda_{l,t} \neq 0, l = 1, \ldots, k-1\}|$ is the number of unassigned measurements.

A standard particle filter (SPF) computes a discrete weighted approximation to the true posterior distribution $p(\boldsymbol{\theta}_t, \boldsymbol{\lambda}_t, \boldsymbol{\alpha}_t \mid \boldsymbol{y}_{1:t})$ using a set of particles

$$p(\boldsymbol{\theta}_t, \boldsymbol{\lambda}_t, \boldsymbol{\alpha}_t \mid \boldsymbol{y}_{1:t}) \approx \sum_{i=1}^{N_{\mathrm{s}}} w_t^{(i)} \delta(\boldsymbol{x}_t^{(i)}, \boldsymbol{\lambda}_t^{(i)}, \boldsymbol{\alpha}_t^{(i)}), \quad (6.13)$$

where $\left\{\boldsymbol{\theta}_t^{(i)}, \boldsymbol{\lambda}_t^{(i)}, \boldsymbol{\alpha}_t^{(i)}\right\}_{i=1}^{N_\mathrm{s}}$ are the samples that characterize the probability distribution $p(\boldsymbol{\theta}, \boldsymbol{\lambda}_t, \boldsymbol{\alpha}_t \mid \boldsymbol{y}_{1:t})$, and $\left\{w_t^{(i)}\right\}_{i=1}^{N_\mathrm{s}}$ are the associated weights. The weight update equation is obtained as [82], [133]

$$\tilde{w}_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(\boldsymbol{y}_t \mid \boldsymbol{\theta}_t^{(i)}, \boldsymbol{\lambda}_t^{(i)}) p(\boldsymbol{\lambda}_t^{(i)}) p(\boldsymbol{\alpha}_t^{(i)} \mid \boldsymbol{y}_{1:t-1}) p(\boldsymbol{x}_t^{(i)} \mid \boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\alpha}_t^{(i)})}{q(\boldsymbol{\theta}_t^{(i)}, \boldsymbol{\lambda}_t^{(i)}, \boldsymbol{\alpha}_t^{(i)} \mid \boldsymbol{\theta}_{t-1}^{(i)}, \boldsymbol{\alpha}_{t-1}^{(i)}, \boldsymbol{y}_{1:t})}, \tag{6.14}$$

with the normalized weights given as

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{i=1}^{N_\mathrm{s}} \tilde{w}_t^{(i)}}. \tag{6.15}$$

The SSPF and the IPPF proposed in [82] combat the curse-of-dimensionality problem by appropriately partitioning the state space. After partitioning, samples of each substate are obtained independently from the corresponding proposal distribution which results in an improved performance. In the following subsections, we extend the SSPF and IPPF architectures to the case of maneuvering targets.

6.3.1 Interacting Multi-Model Sequential Sampling Particle Filtering

The idea of an IMM-SSPF is to sample and update the weight for individual targets in a sequential fashion. The proposal distribution $q(\boldsymbol{\theta}_t, \boldsymbol{\lambda}_t, \boldsymbol{\alpha}_t \mid \boldsymbol{\theta}_{t-1}, \boldsymbol{\alpha}_{t-1}, \boldsymbol{y}_{1:t})$ in the weight update equation given by Eq. (6.14) can be factorized over the individual targets as

$$\begin{aligned} q(\boldsymbol{\theta}_t, \boldsymbol{\lambda}_t, \boldsymbol{\alpha}_t \mid \boldsymbol{\theta}_{t-1}, \boldsymbol{\alpha}_{t-1}, \boldsymbol{y}_{1:t}) \;=\; & \prod_{k=1}^{K} q(\lambda_{k,t} \mid \lambda_{1:k-1,t}, \boldsymbol{\theta}_{k,t}, \boldsymbol{y}_{1:t}) \\ & q(\alpha_{k,t} \mid \alpha_{k-1,t}, \boldsymbol{y}_{1:t}) q(\boldsymbol{\theta}_{k,t}, \mid \boldsymbol{\theta}_{k,t-1}, \alpha_{k,t}, \boldsymbol{y}_{1:t}), \end{aligned} \tag{6.16}$$

where $q(\boldsymbol{\theta}_{k,t}, \mid \boldsymbol{\theta}_{k,t-1}, \alpha_{k,t}, \boldsymbol{y}_{1:t})$ is the state proposal, $q(\alpha_{k,t} \mid \alpha_{k-1,t}, \boldsymbol{y}_{1:t})$ is the kinematic mode proposal, and $q(\lambda_{k,t} \mid \lambda_{1:k-1,t}, \boldsymbol{\theta}_{k,t}, \boldsymbol{y}_{1:t})$ is the association proposal of the k^{th} target, respectively.

State Proposal

The optimal choice for the state proposal distribution $q(\boldsymbol{\theta}_{k,t}, \mid \boldsymbol{\theta}_{k,t-1}, \alpha_{k,t}, \boldsymbol{y}_{1:t})$ of the k^{th} target that minimizes the variance of the associated weights is of the form $p(\boldsymbol{\theta}_{k,t} \mid \boldsymbol{\theta}_{k,t-1}, \alpha_{k,t}, \boldsymbol{y}_{1:t})$ [18], [20]. However sampling from the optimal proposal distribution is not feasible in practice. Therefore, we choose the proposal to be:

$$q(\boldsymbol{\theta}_{k,t}, \mid \boldsymbol{\theta}_{k,t-1}, \alpha_{k,t}, \boldsymbol{y}_{1:t}) = p(\boldsymbol{\theta}_{k,t}, \mid \boldsymbol{\theta}_{k,t-1}, \boldsymbol{\alpha}_{k,t}). \tag{6.17}$$

Kinematic Mode Proposal

We choose the kinematic mode proposal distribution to be $q(\alpha_{k,t} \mid \alpha_{k-1,t}, \boldsymbol{y}_{1:t}) = p(\alpha_{k,t} \mid \boldsymbol{y}_{1:t-1})$. Define $\pi^m_{k,t|t-1} = \Pr(\alpha_{k,t} = m \mid \boldsymbol{y}_{1:t-1})$ (since $\alpha_{k,t}$ is a discrete random variable, we use $\Pr(\cdot)$ to denote its probability distribution from now on.). We can evaluate $\pi^m_{k,t|t-1}$ as

$$\begin{aligned} \pi^m_{k,t|t-1} &= \Pr(\alpha_{k,t} = m \mid \boldsymbol{y}_{1:t-1}) \\ &= \sum_{n=1}^{M} \Pr(\alpha_{k,t} = m \mid \alpha_{k,t-1} = n) \Pr(\alpha_{k,t-1} = n \mid \boldsymbol{y}_{1:t-1}) \\ &= \sum_{n=1}^{M} p_{nm} \pi^n_{k,t-1|t-1}. \end{aligned} \tag{6.18}$$

The update for $\pi^m_{k,t|t}$ is evaluated as

$$\begin{aligned}\pi^m_{k,t|t} &= \Pr(\alpha_{k,t} = m \mid \boldsymbol{y}_{1:t}) \\ &= \frac{p(\boldsymbol{y}_t \mid \alpha_{k,t} = m, \boldsymbol{y}_{1:t-1}) p(\alpha_{k,t} = m \mid \boldsymbol{y}_{1:t-1})}{p(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1})} \\ &\propto p(\boldsymbol{y}_t \mid \alpha_{k,t} = m, \boldsymbol{y}_{1:t-1}) \pi^m_{k,t|t-1}, \end{aligned} \tag{6.19}$$

where

$$\begin{aligned} p(\boldsymbol{y}_t \mid \alpha_{k,t} = m, \boldsymbol{y}_{1:t-1}) &= \int_{\boldsymbol{\theta}_{k,t}} \sum_{\lambda_{k,t}} p(\boldsymbol{y}_t \mid \boldsymbol{\theta}_{k,t}, \lambda_{k,t}, \alpha_{k,t} = m, \boldsymbol{y}_{1:t-1}) p(\boldsymbol{\theta}_{k,t}, \lambda_{k,t} \mid \alpha_t = m, \boldsymbol{y}_{1:t-1}) \mathrm{d}\boldsymbol{\theta}_{k,t} \\ &\approx \sum_{i:\alpha^{(i)}_{k,t}=m} w^{(i)}_{t-1} p\left(\boldsymbol{y}_t \mid \boldsymbol{\theta}^{(i)}_{k,t}, \lambda^{(i)}_{k,t}\right). \end{aligned} \tag{6.20}$$

In the above, the samples $\left\{\boldsymbol{\theta}^{(i)}_{k,t}\right\}_{i=1}^{N_\mathrm{s}}$ are samples drawn from the distribution $p(\boldsymbol{\theta}_{k,t} \mid \boldsymbol{\theta}_{k,t-1}, \alpha_{k,t})$ and $\left\{\lambda^{(i)}_{k,t}\right\}_{i=1}^{N_\mathrm{s}}$ are samples drawn sequentially from the proposal distribution $q(\lambda_{k,t} \mid \lambda_{1:k-1,t}, \boldsymbol{\theta}_{k,t}, \boldsymbol{y}_{1:t})$ described below.

Association Proposal

Here, we describe sampling from the association proposal distribution $q(\lambda_{k,t} \mid \lambda_{1:k-1,t}, \boldsymbol{\theta}_{k,t}, \boldsymbol{y}_{1:t})$ in Eq. (6.16). Sampling the association variables $\lambda_{k,t}$ during filtering, and using these samples to update the weights is called soft-gating procedure [134]. We choose the proposal distribution $q(\lambda_{k,t} = n \mid \lambda_{1:k-1,t}, \boldsymbol{x}_{k,t}, \boldsymbol{y}_t)$ to be of the same form given in [82]

$$q(\lambda_{k,t} = n \mid \lambda_{1:k-1,t}, \boldsymbol{\theta}_{k,t}, \boldsymbol{y}_t) = \begin{cases} \frac{q_{k,0} V^{-1}}{q_{k,0} V^{-1} + \sum_{n=1}^{N_t} q_{k,n} p(\boldsymbol{y}_{n,t} | \boldsymbol{\theta}_{k,t})} & n = 0, \\ \frac{q_{k,n} p(\boldsymbol{y}_{n,t} | \boldsymbol{\theta}_{k,t})}{q_{k,0} V^{-1} + \sum_{n=1}^{N_t} q_{k,n} p(\boldsymbol{y}_{n,t} | \boldsymbol{\theta}_{k,t})} & n \in \{1, 2, \ldots, N_t\}, \end{cases} \tag{6.21}$$

where $q_{k,n} = p(\lambda_k = n \mid \lambda_{1:k-1})$ is the association prior defined in Eq. (6.12) and the likelihood term $p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{k,t})$ is defined below in Eq. (6.23).

The likelihood distribution $p(\boldsymbol{y}_t \mid \boldsymbol{\theta}_t, \boldsymbol{\lambda}_t)$ in Eq. (6.14) is computed as

$$p(\boldsymbol{y}_t \mid \boldsymbol{\theta}_t, \boldsymbol{\lambda}_t) = V^{-N_t^{\mathrm{c}}} \prod_{\substack{k=1 \\ \lambda_{k,t} \in \{1,\ldots,N_t\}}}^{K} p(\boldsymbol{y}_{\lambda_{k,t},t} \mid \boldsymbol{\theta}_{k,t}), \tag{6.22}$$

where

$$p\Big(\boldsymbol{y}_t \mid \boldsymbol{\theta}_{k,t}, \lambda_{k,t}\Big) = p(\boldsymbol{y}_{\lambda_{k,t},t} \mid \boldsymbol{\theta}_{k,t}) = \begin{cases} \frac{1}{V} & \text{if } \lambda_{k,t} = 0, \\ \mathcal{N}\Big(\boldsymbol{y}_{\lambda_{k,t}}; h(\boldsymbol{\theta}_{k,t}), \boldsymbol{\Sigma}_{\mathrm{w}}\Big) & \text{else.} \end{cases} \tag{6.23}$$

Substituting the expressions for the proposal distribution, the likelihood distribution and the prior distribution in the weight update equation given by Eq. (6.14), we get

$$w_t^{(i)} \propto w_{t-1}^{(i)} \left[p(N_t^{\mathrm{c},(i)}) p(N_t^{\mathrm{d},(i)}) V^{-N_t^{\mathrm{c},(i)}}\right] w_{K,t}^{(i)}, \quad \sum_{i=1}^{N_{\mathrm{s}}} w_t^{(i)} = 1, \tag{6.24}$$

where

$$\begin{aligned} w_{k,t}^{(i)} &= w_{k-1,t}^{(i)} \times p(\boldsymbol{y}_{\lambda_{k,t},t} \mid \boldsymbol{\theta}_{k,t}^{(i)}) \times \frac{p(\lambda_{k,t}^{(i)} \mid \lambda_{1:k-1,t}^{(i)})}{q(\lambda_{k,t}^{(i)} \mid \lambda_{1:k-1,t}^{(i)}, \boldsymbol{\theta}_{k,t}^{(i)}, \boldsymbol{y}_t)} \times \frac{p(\alpha_{k,t}^{(i)} \mid \boldsymbol{y}_{1:t-1})}{q(\alpha_{k,t}^{(i)} \mid \boldsymbol{y}_{1:t-1})} \\ &\times \frac{p(\boldsymbol{\theta}_{k,t}^{(i)} \mid \boldsymbol{\theta}_{k,t-1}^{(i)}, \alpha_{k,t}^{(i)})}{q(\boldsymbol{\theta}_{k,t}^{(i)} \mid \boldsymbol{\theta}_{k,t-1}^{(i)}, \alpha_{k,t}^{(i)}, \boldsymbol{y}_{1:t})}, \end{aligned} \tag{6.25}$$

and $N_t^{\mathrm{c},(i)}$, $N_t^{\mathrm{d},(i)}$ are explicitly given by the association vector $\boldsymbol{\lambda}_t^{(i)}$. The representation of the weight update in Eq. (6.24) enables a sampling procedure where the target state, kinematic mode state and the association state are constructed in a sequential fashion for each target. Such a sequential construction enables a resampling steps for

the individual targets when the variance of the weights associated with the samples of the individual targets, i.e., $w_{k,t}^{(i)}$ becomes high. However, this kind of sampling is sensitive to the order in which the individual targets are sampled. In order to overcome this dependence on the sampling order, the authors in [82] suggest repeating this procedure a number of times for different orderings of the targets, and then obtaining the final Monte-Carlo representation by combining the samples obtained from the individual runs. The overall algorithm for the IMM-SSPF is given in Algorithm 5.

Algorithm 5 IMM-SSPF Algorithm for MTT with Association Ambiguities.

1: Initialize set of particles $\{\boldsymbol{\theta}_{k,0}\}_{i=1}^{N_\mathrm{s}}$, the weights $w_{k,0}^{(i)} = 1/N_\mathrm{s}$ and the target mode probabilities $\boldsymbol{\pi}_{k,1|0}$ for $k = 1, \ldots, K$.
2: **for** $t = 1 : T$ **do**
3: **for** $k = 1 : K$ **do**
4: Draw samples $\{\boldsymbol{\alpha}_{k,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ from the kinematic mode proposal given by Eq. (6.18).
5: Draw samples $\{\boldsymbol{\theta}_{k,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ from the state proposal given by Eq. (6.17).
6: Draw samples $\{\lambda_{k,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ from the association proposal given by Eq. (6.21).
7: Update the weights $\{w_{k,t}^{(i)}\}_{i=1^{N_\mathrm{s}}}$ following Eq. (6.25) and normalize the weights such that $\sum_{i=1}^{N_\mathrm{s}} w_{k,t}^{(i)} = 1$.
8: Update the mode probabilities using Eq. (6.19), and Eq. (6.20).
9: Depending on the weights $w_{k,t}^{(i)}$, resample the set $\{\boldsymbol{\theta}_{k,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ to obtain a new particle set.
10: **end for**
11: Compute the overall weights using Eq. (6.24) and normalize $\sum_{i=1}^{N_\mathrm{s}} w_t^{(i)} = 1$.
12: Obtain the estimates using $\hat{\boldsymbol{\theta}}_t = \sum_{i=1}^{N_\mathrm{s}} w_t^{(i)} \boldsymbol{\theta}_t^{(i)}$ and resample if necessary.
13: **end for**

6.3.2 Interacting Multi-Model Independent Partition Particle Filtering

In an IMM-IPPF, the assumption A2 is replaced by A3, i.e., the target associations are assumed to be independent of each other. The independence assumption is achieved by choosing the prior to be

$$p(\lambda_{k,t} = n \mid \lambda_{1:k-1,t}) = p(\lambda_{k,t} = n) \propto \begin{cases} 1 - P_{\mathrm{d}} & n = 0, \\ \frac{P_{\mathrm{d}}}{K} & \text{else.} \end{cases} \tag{6.26}$$

As a result of this choice, the proposal distribution corresponding to individual target associations will become independent of each other, and hence an efficient target-wise sampling and weight updating procedure can be used. The weights for the k^{th} target are updated using

$$w_{k,t}^{(i)} \propto p(\boldsymbol{y}_{\lambda_{k,t},t} \mid \boldsymbol{\theta}_{k,t}^{(i)}) \times \frac{p(\lambda_{k,t}^{(i)})}{q(\lambda_{k,t}^{(i)} \mid \boldsymbol{\theta}_{k,t}^{(i)}, \boldsymbol{y}_t)} \times \frac{p(\alpha_{k,t}^{(i)} \mid \boldsymbol{y}_{1:t-1})}{q(\alpha_{k,t}^{(i)} \mid \boldsymbol{y}_{1:t-1})} \times \frac{p(\boldsymbol{\theta}_{k,t}^{(i)} \mid \boldsymbol{\theta}_{k,t-1}^{(i)}, \alpha_{k,t}^{(i)})}{q(\boldsymbol{\theta}_{k,t}^{(i)} \mid \boldsymbol{\theta}_{k,t-1}^{(i)}, \boldsymbol{\alpha}_{k,t}^{(i)}, \boldsymbol{y}_{1:t})}, \tag{6.27}$$

where the association proposal $q(\lambda_{k,t} \mid \boldsymbol{\theta}_{k,t}, \boldsymbol{y}_t)$ is obtained following Eq. (6.21) and is given as

$$q(\lambda_{k,t} \mid \boldsymbol{\theta}_{k,t}, \boldsymbol{y}_t) \begin{cases} \frac{q_{k,0} V^{-1}}{q_{k,0} V^{-1} + \sum_{n=1}^{N_t} q_{k,n} p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{k,t})} & n = 0, \\ \frac{q_{k,n} p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{k,t})}{q_{k,0} V^{-1} + \sum_{n=1}^{N_t} q_{k,n} p(\boldsymbol{y}_{n,t} \mid \boldsymbol{\theta}_{k,t})} & n \in \{1, 2, \ldots, N_t\}, \end{cases} \tag{6.28}$$

where $q_{k,n} = p(\lambda_k = n)$ is the association prior defined in Eq. (6.26). The overall weight update equations are given as

$$w_t^{(i)} \propto w_{t-1}^{(i)} \left[p(N_t^{\mathrm{c},(i)}) p(N_t^{\mathrm{d},(i)}) V^{-N_t^{\mathrm{c},(i)}} \right] \prod_{k=1}^{K} w_{k,t}^{(i)}, \quad \sum_{i=1}^{N_\mathrm{s}} w_t^{(i)} = 1. \tag{6.29}$$

We call this filtering the IMM-IPPF. Since the association variable of each target is obtained independent of each other, the inner loop in IMM-IPPF can be implemented in a parallel fashion, and this implementation leads to a lower computational time. However, the independence assumption A3 used in IMM-IPPF is not realistic when the probability of detection is low and the clutter density is high. Hence, the filter performance degrades significantly in such scenarios. The Algorithm is summarized 6.

6.4 Game theory Preliminaries

Game theory [135], [136] is a popular tool for distributed control in multi-agent systems. Using game theory for distributed control requires modeling the interactions between the multiple agents. The central component of this modeling is identifying the agents themselves, identifying the strategy space for the agents, and assigning a utility function to each agent. The goal is to design utility functions such that the resulting game possess desirable properties.

Algorithm 6 IMM-IPPF Algorithm for MTT with Association Ambiguities.

Initialize set of particles $\{\boldsymbol{\theta}_{k,0}\}_{i=1}^{N_\mathrm{s}}$, the weights $w_{k,0}^{(i)} = 1/N_\mathrm{s}$ and the target mode probabilities $\boldsymbol{\pi}_{k,1|0}$ for $k = 1, \ldots, K$.
for $t = 1 : T$ **do**
  parfor $k = 1 : K$ **do**
    Draw samples $\{\boldsymbol{\alpha}_{k,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ from the kinematic mode proposal given by Eq. (6.18).
    Draw samples $\{\boldsymbol{\theta}_{k,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ from the state proposal given by Eq. (6.17).
    Draw samples $\{\lambda_{k,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ from the association proposal given by Eq. (6.28).
    Update the weights $\{w_{k,t}^{(i)}\}_{i=1^{N_\mathrm{s}}}$ following Eq. (6.27) and normalize the weights such that $\sum_{i=1}^{N_\mathrm{s}} w_{k,t}^{(i)} = 1$.
    Update the mode probabilities using Eq. (6.19), and Eq. (6.20).
    Resample the set $\{\boldsymbol{\theta}_{k,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ according to $w_{k,t}^{(i)}$, to obtain a new particle set, and reinitialize the weights as $w_{k,t}^{(i)} = 1/N,\ \forall i$.
  end for
  Compute the overall weights using

$$w_t^{(i)} \propto w_{t-1}^{(i)} \left[p(N_t^{\mathrm{c},(i)}) p(N_t^{\mathrm{d},(i)}) V^{-N_t^{\mathrm{c},(i)}} \right]$$

  and normalize $\sum_{i=1}^{N_\mathrm{s}} w_t^{(i)} = 1$.
  Obtain the estimates using $\hat{\boldsymbol{\theta}}_t = \sum_{i=1}^{N_\mathrm{s}} w_t^{(i)} \boldsymbol{\theta}_t^{(i)}$ and resample if necessary.
end for

6.4.1 Normal Form Game

A normal (or a strategic) form game Λ consists of a set $\Omega = \{1, 2, \ldots, N\}$ of N players, where each player $n \in \Omega$ has a strategy set $\mathcal{S}_n$ and a utility function $u_n : \mathcal{S} \to \mathbb{R}$, with $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \mathcal{S}_N$ being the set of strategy profiles. For a strategy $s = (s_1, \ldots, s_N) \in \mathcal{S}$, we write s_{-n} to denote the profile of all player strategies other than the player n, i.e.,

$$s_{-n} = (s_1, \ldots, s_{n-1}, s_{n+1}, \ldots, s_N). \tag{6.30}$$

With this notation, we sometimes represent the utility $u_n(s)$ as $u_n(s_n, s_{-n})$.

6.4.2 Nash Equilibrium

Nash Equilibrium (NE) is a well-known equilibrium concept that emerges in noncooperative games. A strategy profile $s^* = (s_1^*, \ldots, s_N^*)$ is called a pure strategy Nash equilibrium iff

$$u_n(s_n^*, s_{-n}^*) \geq u_n(s_n, s_{-n}^*), \quad \text{for any } s_n \in \mathcal{S}_n \text{ and } \forall n \in \Omega. \tag{6.31}$$

Computing a NE, given a normal form game is a fundamental problem in game theory. The problem is combinatorial, and it belongs to a class of complex algorithms called PPAD [137] [14]. It should also be noted that the NE does not always lead to the best performance in a distributed, multiplayer game when the players are coordinating. Correlated Equilibrium (CE) is a generalization of NE that considers the ability of

[14] PPAD stands for "polynomial party argument"; see [138] for a formal definition and examples of other PPAD problems. It is believed that PPAD-complete problems are not solvable in polynomial time, but are simpler than NP-complete problems, although this remains an open problem [137].

	Stop	Go
Stop	(4,4)	(1,5)
Go	(5,1)	(0,0)

Table 6.1: Utilites of the players for the traffic signal game

players to coordinate actions. Hence, CE leads to a higher degree of cooperation and a better solution compared to the non-cooperative NE.

6.4.3 Correlated Equilibrium

Consider a simple traffic game with two players, where each player has two strategies: 'stop' or 'go'. The utilites obtained by these players are tabulated as shown in Table 6.1, where the row corresponds to the strategy of the first player and the column corresponds to the strategy of the second player.

It can be seen that this game has two pure strategy Nash equilibria given by (Stop,Go) and (Go,Stop), and one mixed strategy Nash equilibrium where each player chooses one of the strategies with a probability 1/2. The utility of each player by playing the mixed strategy equilibrium can be shown to be 2.5. Now, suppose that there is a traffic signal installed that tells one of the players to stop and other player to go. Such a fair random signal recommending actions to each player will result in a new equilibrium that eliminates the lower payoff outcomes. This kind of an equilibrium, where a trusted source provides a 'private' recommendation to each player is called as a correlated equilibrium (CE) [139]. In a CE, there is no incentive for either player to deviate from the recommendation given by the signal. Further, it can be shown that the utility obtained by the players using the CE will be higher (in this example, the utility using the CE will be 3) than the utility obtained from the Nash

equilibrium. For a general game, if the source draws a strategy profile s from a probability distribution π and announces to each player n separately (and privately) n's own component, then the player will have no incentive to choose another strategy, assuming that the other players also conform to the recommendation provided by the source. Finally note that the strategies of the players in a CE are not independent of each other, as they depend on a commonly observed signal.

Mathematically, a probability distribution π is called a correlated equilibrium if for all players $n \in \Omega$ and all strategies $i, j \in \mathcal{S}_n$, we have

$$\sum_{s \in \mathcal{S}: s_n = i} \pi(s) \Big[u_n(j, s_{-n}) - u_n(s) \Big] \leq 0. \tag{6.32}$$

It can be seen that a CE is a generalization of the Nash Equilibrium. Every Nash Equilibrium is a CE. In fact, Nash equilibria correspond to the special case where π is a product measure, i.e., the players select the actions independently. In this case, the recommendations provided are (stochastically) independent across the players.

The CE has several important advantages. The set of correlated equilibria is nonempty, closed and convex [139] They are simple to understand, guaranteed to exist, and computationally feasible (CE can be found in polynomial time for any number of players and strategies by linear programming [140]).

6.5 Concurrent Data Association via Game Theory

In this section, we will solve the data association problem by formulating it as a game. We will find the equilibrium of the game using regret matching learning algorithm, which will produce an association vector. Using regret matching, each target can be associated to the measurement simultaneously, thereby enabling a parallel implementation. Hence, we call our approach as concurrent. We will first describe the game, and then talk about the equilibrium.

6.5.1 Data Association Game

Consider a game with a set Ω of K trackers as the players. Each player wants to track a particular target, which is assumed to be known to the player and remains unchanged throughout the duration. The strategy set $\mathcal{S}_k = \{0, 1, 2, \ldots, N_t\}$ of each player corresponds to the set of measurements that are known to all of the players, and the strategy $s_k \in \mathcal{S}_k$ allows the player to choose one measurement from the set of all measurements or 0 in case the corresponding target does not produce any measurement. For example, $s_k = 1$, implies that the k^{th} player uses measurement $\boldsymbol{y}_1$ to update the weights corresponding to the k^{th} target. For each player, we define a utility function $u_k(s_k, s_{-k}) : \mathcal{S} \rightarrow \mathbb{R}$, with $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_K$ in the next subsection. We use $\mathcal{U}$ to denote the set of utility functions of all the players, i.e., $\mathcal{U} = \{u_k\}_{k=1}^{K}$. The game $\Gamma\Big(\Omega, \mathcal{S}, \mathcal{U}\Big)$ defined by the set of players Ω, the strategy set $\mathcal{S}$, and the utility functions $\mathcal{U}$ is called the data association game.

6.5.2 Utility Functions

For the k^{th} player, we define the utility function as follows:

$$u_k\left(s_k, s_{-k}\right) = \begin{cases} d_k(s_k) + \mu_1 g_k(s_k, s_{-k}) & s_k \neq 0, \\ \mu_2 & s_k = 0, \end{cases} \tag{6.33}$$

where

$$d_k(s_k) = -\frac{1}{K-1}\left[\left(\boldsymbol{y}_{s_k} - f(\tilde{\boldsymbol{\theta}}_k)\right)^T \boldsymbol{\Sigma}_{\mathrm{w}}^{-1}\left(\boldsymbol{y}_{s_k} - f(\tilde{\boldsymbol{\theta}}_k)\right) - d_{\max}\right] \tag{6.34}$$

is the scaled Mahalonobis distance between the true measurement $\boldsymbol{y}_{s_k}$ and the predicted measurement $f(\tilde{\boldsymbol{\theta}}_k)$, and the function $g_k(s_k, s_{-k})$ is defined as

$$g_k(s_k, s_{-k}) = \frac{1}{K-1}\sum_{\substack{l=1 \\ l\neq k}}^{K} \|s_k - s_l\|_{\ell_0}. \tag{6.35}$$

In Eq. (6.33) above, $\mu_1 > 0, \mu_2 > 0$ are fixed constants, and $d_{\max}$ denotes half the maximum Mahalonobis distance between the actual measurement and the predicted measurement. Note that the scaled Mahalonobis distance $d_k(s_k)$ is bounded as $-d_{\max} \leq (K-1)d_k(s_k) \leq d_{\max}$, $\forall s_k$, and $g_k(s_k, s_{-k}) \leq 1$, $\forall s \in \mathcal{S}$.

The function in Eq. (36) models the utility that the k^{th} target derives by choosing a measurement s_k from the set of all the available measurements, as function of the measurement s_k itself and the measurements chosen by other targets, which we denoted using s_{-k}. The utility function when $s_k \neq 0$, i.e., when the k^{th} target produces

a measurement, has two terms:

$$u_k\left(s_k, s_{-k}\right) = \underbrace{d_k(s_k)}_{\text{first term}} + \underbrace{\mu_1 g_k(s_k, s_{-k})}_{\text{second term}} \tag{6.36}$$

The first term corresponds to the negative of (scaled) Mahalonobis distance between the true measurement and the predicted measurement. It is easy to observe that this term does not depend on other targets' measurement choice. The larger this term, the higher the probability that the k^{th} target actually produces the measurement $\boldsymbol{y}_{s_k}$. The second term quantifies the feasibility of the measurement that a particular target chooses. If none of the other targets choose the measurement $\boldsymbol{y}_{s_k}$, then the second term takes its maximum value (which is unity for our case). If one of the other targets chooses the same measurement $\boldsymbol{y}_{s_k}$, then the value of the second term decreases. Hence, the overall utility decreases as well. Thus, for the utility to be high, each target chooses a measurement that (i) maximizes the negative of the distance (or equivalently minimize the distance) between the true measurement and the predicted measurement; and (ii) none of the other targets choose.

6.5.3 Correlated Equilibrium using Regret Matching

In this section, we will describe how a correlated equilibrium of the data association game can be reached using a learning mechanism. First, the existence of CE for the data association game is guaranteed using the following Lemma.

Lemma 1. *For the data association game* $\Gamma\Big(\Omega, \mathcal{S}, \mathcal{U}\Big)$*, a CE always exists.*

Proof. Theorem 1 of [141] proves that every finite game has a non-empty set of correlated equilibria. It is obvious that $\Gamma\Big(\Omega, \mathcal{S}, \mathcal{U}\Big)$ is a finite game. Hence, Theorem 1 applies directly, and a CE exists. □

We use an iterative learning algorithm called the regret matching [142] to find the CE. Let $s_k(j)$ denote the strategy of the k^{th} player in the j^{th} iteration. Note that $s_k(j) \in \{0, 1, 2, \ldots, N_t\}$, where N_t is the number of strategies. Each player computes the average regret for choosing the n^{th} strategy for $n \in \{0, 1, 2, \ldots, N_t\}$ in the j^{th} iteration

$$\begin{aligned} r^n(j) &= \max\{R^n(j), 0\} && (6.37) \\ R_k^n(j) &= \frac{1}{j-1}\sum_{l=1}^{j-1}\Big[u_k\big(n, s_{-k}(l)\big) - u_k\big(s(l)\big)\Big]. && (6.38) \end{aligned}$$

Lemma 2. *The k^{th} player can recursively compute the n^{th} component of $\boldsymbol{R}_k$ as*

$$R_k^n(j) = \left(\frac{j-2}{j-1}\right) R_k^n(j-1) + \frac{1}{j-1}\left[u_k\Big(n, s_{-k}(j-1)\Big) - u_k\Big(s(j-1)\Big)\right]. \quad (6.39)$$

Proof. Consider

$$\begin{aligned} R_k^n(j) &= \frac{1}{j-1}\sum_{l=1}^{j-1}\left[u_k\Big(n, s_{-k}(l)\Big) - u_k\Big(s(l)\Big)\right] \\ &= \frac{1}{j-1}\sum_{l=1}^{j-2}\left[u_k\Big(n, s_{-k}(l)\Big) - u_k\Big(s(l)\Big)\right] + \frac{1}{j-1}\left[u_k\Big(n, s_{-k}(j-1)\Big) - u_k\Big(s(j-1)\Big)\right] \\ &= \left(\frac{j-2}{j-1}\right) R_k^n(j-1) + \frac{1}{j-1}\left[u_k\Big(n, s_{-k}(j-1)\Big) - u_k\Big(s(j-1)\Big)\right]. \quad (6.40) \end{aligned}$$

□

The expression $r_k^n(j)$ has an interpretation as the measure of the average regret at time j for not having played the strategy n up to time j. Let p_k^n denote the probability that the k^{th} player chooses n^{th} strategy. In regret matching, each player chooses a strategy according to the distribution $\boldsymbol{p}_k(j)$ which is proportional to the regret vector of the player. Let $s(j-1) = (l, s_{-k}(j-1))$. Then $\forall\ k$

$$p_k^n(j) = \begin{cases} \frac{1}{\mu} r_k^n(j) & \text{if } l \neq n \\ 1 - \sum_{\substack{n \in \{0,1,\ldots,N_t\} \\ n \neq l}} p_k^n(j) & l = n. \end{cases} \tag{6.41}$$

The constant $\mu > 0$ is fixed throughout the procedure and is generally chosen to be $\mu > 2\max(d_{\max} + \mu_1, \mu_2)(N_t + 1)$ [142]. This choice of μ guarantees that there is always a positive probability of playing same strategy as in the previous step. We give below two results from [142] (see [142] for proofs) that characterize the dynamics of regret-based learning.

Lemma 3. *If every player k plays according to the adaptive distributions given by Eq. (6.41), then the empirical distributions of play converge almost surely as $t \to \infty$ to the set of correlated equilibrium.*

Lemma 4. *Let $s(j) = \big(l, s_{-k}(j)\big), j \in \mathbb{N}$ be a sequence of plays of the k^{th} player and let $\epsilon \geq 0$. Then $\limsup_{t\to\infty} r_k^n(j) \leq \epsilon$ for every $k \in \Omega$ and $n \in \{0, 1, \ldots, N_t\}, n \neq l$ if and only if the empirical distributions $\pi_j(s)$ defined as $\pi_j(s) = \frac{1}{j}|\{\tau \leq j : s(\tau) = s\}|$ converge to set of correlated equilibria.*

It can be seen that in regret matching the correlation in the plays of different players arises from the commonly observed history. Thus, the history serves as a signal in giving the private recommendation to each player. Further, since the update for $R_n^k(j)$ requires the associations of other targets at $(j-1)^{th}$ iteration, only the association

vector corresponding to $(j-1)^{th}$ iteration, i.e., $s(j-1)$ has to be communicated to each of the trackers.

6.5.4 Equilibrium Characterization of Data Association Game

All the results we stated in the above subsection hold for any finite game. In this subsection, we will characterize the equilibrium points that are obtained via regret based learning for a data association game through a series of Lemmas. Following Lemma 4, observe that the empirical distributions $\boldsymbol{p}(j)$ converge to pure equilibrium if it exists, or they must be infinitely often outside the set of correlated equilibria, since after correlated equilibrium is reached the play doesn't change as the regrets converge to zero (see also [142]). We are interested in the case when the empirical distributions $\boldsymbol{p}_k(j)$ converge to pure equilibrium, $\forall\ k$. We show using numerical simulations that this will be the case for data association games.

We now characterize the feasibility of the equilibrium obtained using regret matching in data association games. We provide sufficient conditions for the equilibrium to be a feasible association vector.

Lemma 5. *If s^* is a pure strategy equilibrium of the data association game $\Gamma\Big(\Omega, \mathcal{S}, \mathcal{U}\Big)$ and $\mu_1 > 2d_{\max}$, then s^* is a feasible association vector.*

Proof. Since s^* is a pure correlated equilibrium, it is necessarily a Nash equilibrium. We prove the lemma using contradiction. Suppose that the equilibrium s^* is not feasible, i.e, there exists at least one pair l, k, such that $l \neq k$, $s_k^* = s_l^*$ and $s_k^*, s_l^* \neq 0$. We will construct a strategy s' which is feasible in the following manner. First, we consider the case where $N_t > K$. Let $s^* = (s_k^*, s_{-k}^*)$. Construct $s' = (s_k', s_{-k}^*)$ such

that $s'_k \neq s^*_l$ $l = 1, 2, \ldots, K$, i.e., expect for the k^{th} component of the strategy vector, s' is same as s^*. We select the k^{th} component of s' such that it is a feasible vector. Since $N_t > K$, we can find at least one such construction[15]. For the k^{th} player, we have

$$\begin{aligned} u_k(s^*_k, s^*_{-k}) - u_k(s'_k, s^*_{-k}) &= d_k(s^*_k) + \mu_1 g_k(s^*_k, s^*_{-k}) - d_k(s'_k) - \mu_1 g_k(s'_k, s^*_{-k}) \\ &= \Big(d_k(s^*_k) - d_k(s'_k)\Big) + \frac{\mu_1}{K-1}\Big(\sum_{\substack{l=1 \\ l\neq k}}^{K} \|s^*_k - s^*_l\|_{\ell_0} - \sum_{\substack{l=1 \\ l\neq k}}^{K} \|s'_k - s^*_l\|_{\ell_0}\Big) \\ &\leq \frac{1}{K-1}\Big[2d_{\max} + \mu_1\Big[(K-2) - (K-1)\Big]\Big] \\ &= \frac{1}{K-1}\left[2d_{\max} - \mu_1\right] \\ &< 0. \end{aligned} \tag{6.42}$$

The inequality implies that (s^*_k, s^*_{-k}) is not a NE. If $N_t < K$, then we construct s' such that the $s'_k = 0$, $s'_l = s^*_l$, $l \neq k$. Following exactly the same argument as above, it can be shown that s^* is not a NE for this case also. Thus, our assumption on the feasibility of the NE is wrong and s^* is feasible. □

Next, we provide necessary and sufficient conditions in terms of the parameters μ_1 and μ_2 for the regret matching algorithm to incorrectly associate the measurements. We consider two types of errors: the false alarms and the missed detections, which we define below.

Definition 3. *A* false alarm *occurs if for any k, $\lambda^{\text{act}}_k = 0$ and $\lambda^{\text{est}}_k = m$ where $m > 0$.*

[15]In a similar way, we can construct a strategy vector for the cases where there is more than one location where the strategies are the same, i.e., $s^*_k = s^*_l$ for more than one l. The proof for this scenario follows the similar steps as above. However, for this scenario the new vector will not be feasible anymore. But the proof steps will remain the same.

Definition 4. *A* missed detection *occurs if for any* k, $\lambda_k^{\text{act}} = m$ *and* $\lambda_k^{\text{est}} = 0$ *where* $m > 0$.

In the above, λ_k^{act} and λ_k^{est} are the actual and the estimated associations for the k^{th} target respectively.

Lemma 6. *Assume that* $\lambda_k^{\text{act}} = 0$ *for some target* $k \in \Omega$. *Let* λ_k^{est} *be the estimated association for the* k^{th} *target. Then the necessary and sufficient condition for* $\lambda_k^{\text{est}} = m$, *where* $m \neq 0$ *is* $d_k(m) \geq \mu_2 - \mu_1$.

Proof. We know that the association vector $\boldsymbol{\lambda}^{\text{est}}$ is a pure correlated equilibrium of the data association game. Therefore $\boldsymbol{\lambda}^{\text{est}}$ should necessarily be a NE. Using this fact, and that $\boldsymbol{\lambda}^{\text{est}}$ is feasible, we have a necessary and sufficient condition

$$\begin{aligned}
& u_k(n, s_{-k}) - u_k(m, s_{-k}) \leq 0 \ \forall \ n \neq m \\
& \Leftrightarrow u_k(0, s_{-k}) - u_k(m, s_{-k}) \leq 0 \\
& \Leftrightarrow \mu_2 - \Big(d_k(m) + \mu_1 g_k(m, s_{-k}) \Big) \leq 0 \\
& \Leftrightarrow \mu_2 - d_k(m) - \mu_1 \leq 0 \\
& \Leftrightarrow d_k(m) \geq \mu_2 - \mu_1
\end{aligned} \tag{6.43}$$

□

It can be seen that regret matching assigns any clutter measurement m satisfying $d_k(m) \geq \mu_2 - \mu_1$ to targets that do not produce measurements, and thereby produces false alarms.

Lemma 7. *Assume that* $\lambda_k^{\text{act}} = m \neq 0$ *for a target* $k \in \Omega$. *Let* λ_k^{est} *be the estimated association for the* k^{th} *target. Then the necessary and sufficient condition for* $\lambda_k^{\text{est}} = 0$, *is* $d_k(m) \leq \mu_2 - \mu_1$.

Proof. Since λ_k^{est} is a pure correlated equilibrium and a feasible association vector, we have that the necessary and sufficient condition for $\lambda_k^{\text{est}} = 0$ when $\lambda_k^{\text{act}} \neq 0$ is

$$\begin{aligned} & u_k(0, s_{-k}) - u_k(m, s_{-k}) \geq 0 \\ \Leftrightarrow\ & \mu_2 - \Big(d_k(m) + \mu_1 g_k(m, s_{-k})\Big) \geq 0 \\ \Leftrightarrow\ & \mu_2 - d_k(m) - \mu_1 \geq 0 \\ \Leftrightarrow\ & d_k(m) \leq \mu_2 - \mu_1 \end{aligned} \tag{6.44}$$

□

It can be seen now that when a target measurement m is such that $d_k(m) \leq \mu_2 - \mu_1$, then it will be wrongly associated to as a missed detection, and hence produces missed detections. Lemmas 6 and 7 provide a way to choose the parameter μ_2 for the utility functions. If we choose μ_2 such that $\mu_2 - \mu_1$ is small, then for any noisy target originated measurement m from the k^{th} target, the probability $\Pr[d_k(m) \geq \mu_2 - \mu_1]$ will be high. As a result, k^{th} target will be correctly associated to the actual measurement decreasing the probability of a miss. However, for a clutter originated measurement l the probability $\Pr[d_k(l) \geq \mu_2 - \mu_1]$ is high if $\mu_2 - \mu_1$ is small. Therefore, every time the k^{th} target does not produce a measurement, it will be associated to the l^{th} clutter, increasing the probability of false alarm. On the other hand if μ_2 is chosen such that $\mu_2 - \mu_1$ is large then, for target originated measurement m, $\Pr[d_k(m) \leq \mu_2 - \mu_1]$ is high and hence the probability of a miss increases. But for a

clutter originated measurement m, $\Pr[d_k(m) \geq \mu_2 - \mu_1]$ is low and hence probability of false alarms decreases. In this manner, the choice of the parameters μ_2 and μ_1 determine the trade off between the probability of false alarms and the probability of miss.

6.6 Interacting Multi-Model Concurrent Particle Filtering

In this section, we propose a concurrent filtering (IMM-CPF) method that combines the advantages of both IMM-SSPF and IMM-IPPF, i.e., we develop a particle filtering technique that enables a parallel implementation without using the independence assumption A3. In IMM-CPF we first deterministically find the best association between the targets and the measurements by formulating the data association as a game, as discussed in the previous section. We then find the equilibrium of the game using regret matching algorithm in a concurrent manner. The equilibrium will produce a feasible association vector, which we use for all the samples of the target state and the kinematic mode state. Further, once the association is known, the joint filtering distribution factorizes completely over the individual targets. Each of the targets is then independent of each other and thus, K particle filters operating in a parallel fashion are used to obtain the estimates of the individual target states. The weight update equation corresponding to the k^{th} particle filter is given as

$$w_{k,t}^{(i)} = p(\boldsymbol{y}_{\lambda_{k,t},t} \mid \boldsymbol{\theta}_{k,t}^{(i)}) \times \frac{p(\alpha_{k,t}^{(i)} \mid \boldsymbol{y}_{1:t-1})}{q(\alpha_{k,t}^{(i)} \mid \boldsymbol{y}_{1:t-1})} \frac{p(\boldsymbol{\theta}_{k,t}^{(i)} \mid \boldsymbol{\theta}_{k,t-1}, \alpha_{k,t}^{(i)})}{q(\boldsymbol{\theta}_{k,t}^{(i)} \mid \boldsymbol{\theta}_{k,t-1}^{(i)}, \alpha_{k,t}^{(i)}, \boldsymbol{y}_{1:t})}. \tag{6.45}$$

Since the association vector is same for all the particles, IMM-CPF avoids the computation of the overall weights and the state corresponding to each target can be found independently. As a result, we can implement tracking filters for all the targets simultaneously. The overall Algorithm is summarized in 7.

Algorithm 7 IMM-CPF Algorithm for MTT with Association Ambiguities.

Initialize set of particles $\{\boldsymbol{\theta}_{k,0}\}_{i=1}^{N_\mathrm{s}}$, the weights $w_{k,0}^{(i)} = 1/N_\mathrm{s}$ and the target mode probabilities $\boldsymbol{\pi}_{k,1|0}$ for $k = 1, \ldots, K$.
for $t = 1 : T$ **do**
  Find the association vector $\boldsymbol{\lambda}$ using regret matching in a *distributed manner*.
  parfor $k = 1 : K$ **do**
    Draw samples $\{\boldsymbol{\alpha}_{k,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ from the kinematic mode proposal given by Eq. (6.18).
    Draw samples $\{\boldsymbol{\theta}_{k,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ from the state proposal given by Eq. (6.17).
    Update the weights $\{w_{k,t}^{(i)}\}_{i=1^{N_\mathrm{s}}}$ following Eq. (6.45) and normalize the weights such that $\sum_{i=1}^{N_\mathrm{s}} w_{k,t}^{(i)} = 1$.
    Update the mode probabilities using Eq. (6.19), and Eq. (6.20).
    Resample the set $\{\boldsymbol{\theta}_{k,t}^{(i)}\}_{i=1}^{N_\mathrm{s}}$ according to $w_{k,t}^{(i)}$, to obtain a new particle set.
    Obtain the estimates using $\hat{\boldsymbol{\theta}}_{k,t} = \frac{1}{N_\mathrm{s}} \sum_{i=1}^{N_\mathrm{s}} \boldsymbol{\theta}_{k,t}^{(i)}$.
  end for
end for

6.7 Numerical Results

In this section, we use numerical examples to demonstrate the performance of the tracking filters. We consider three tracking scenarios with varying complexity and evaluate the performance of the four tracking filters for these scenarios. We will describe the simulation setup first and then discuss the results.

We consider surveillance of a region for a period of 30 tracking intervals. The duration of each tracking interval was 0.1 s ($\Delta t = 0.1s$). We assume that there is one sensor

located at the origin of the coordinated system $(0, 0)$. Each measurement $\boldsymbol{y}_{n,t} = [y_{1,n,t}, y_{2,n,t}, y_{3,n,t}]^T$ is a 3-dimensional vector with

$$y_{1,n,t} = \sqrt{\rho^2_{\mathrm{x},k,t} + \rho^2_{\mathrm{y},k,t}} + w_{1,n,t}, \tag{6.46}$$

$$y_{2,n,t} = \tan^{-1}(\frac{\rho_{\mathrm{y},k,t}}{\rho_{\mathrm{x},k,t}}) + w_{2,n,t} \text{ and,} \tag{6.47}$$

$$y_{3,n,t} = \frac{\dot{\rho}_{\mathrm{x},k,t}\rho_{\mathrm{x},k,t} + \dot{\rho}_{\mathrm{y},k,t}\rho_{\mathrm{y},k,t}}{\sqrt{\rho^2_{\mathrm{x},k,t} + \rho^2_{\mathrm{y},k,t}}} + w_{3,n,t} \tag{6.48}$$

if the n^{th} measurement originates from the k^{th} target and $\boldsymbol{\theta}_{k,t} = [\rho_{\mathrm{x},k,t}, \rho_{\mathrm{y},k,t}, \dot{\rho}_{\mathrm{x},k,t}, \dot{\rho}_{\mathrm{y},k,t}]^T$ is the state of the k^{th} target. Equations (6.46), (6.47) and (6.48) together constitute the measurement model given by Eq. (6.4). The covariance matrix $\boldsymbol{\Sigma}_{\mathrm{w}} = \mathrm{diag}\{\sigma^2_{\mathrm{w},1}, \sigma^2_{\mathrm{w},2}, \sigma^2_{\mathrm{w},3}\}$ is a diagonal matrix with $\sigma^2_{\mathrm{w},1} = 100$, $\sigma^2_{\mathrm{w},2} = 1 \times 10^{-4}$ and $\sigma^2_{\mathrm{w},3} = 1 \times 10^{-2}$. We considered tracking under three scenarios: easy, medium and hard. For each of these scenarios, we chose the probability of detection P_{d} and the clutter density γ/V (see Eq. (6.6)) as

$$\text{Easy} \;\; P_{\mathrm{d}} = 1, \quad \gamma/V = 0.5$$

$$\text{Medium} \quad P_{\mathrm{d}} = 0.8, \gamma/V = 3$$

$$\text{Hard} \;\; P_{\mathrm{d}} = 0.6, \gamma/V = 5$$

We assume that there are three targets ($K = 3$) moving in the region of interest, and each of the these targets to have three kinematic modes ($M = 3$). For each target, the state transition equation given by Eq. (6.1) takes the form

$$\boldsymbol{\theta}_{k,t} = \boldsymbol{F}_{\alpha_{k,t}}\boldsymbol{\theta}_{k,t-1} + \boldsymbol{v}_{k,t-1}, \tag{6.49}$$

where $\boldsymbol{F}_{\alpha_{k,t}}$ is the state transition matrix corresponding to the mode $\alpha_{k,t}$. In this work, we choose the first mode to be a constant velocity model with

$$\boldsymbol{F}_1 = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and the second and third modes to be co-ordinated turn models with

$$\boldsymbol{F}_2 = \begin{bmatrix} 1 & 0 & \frac{\sin(\pi\Delta t)}{\Delta t} & \frac{1-\cos(\pi\Delta t)}{\Delta t} \\ 0 & 1 & \frac{-(1-\cos(\pi\Delta t))}{\Delta t} & \frac{\sin(\pi\Delta t)}{\Delta t} \\ 0 & 0 & \frac{\cos(\pi\Delta t)}{\Delta t} & \frac{-\sin(\pi\Delta t)}{\Delta t} \\ 0 & 0 & \frac{\sin(\pi\Delta t)}{\Delta t} & \frac{\cos(\pi\Delta t)}{\Delta t} \end{bmatrix} \text{ and } \boldsymbol{F}_3 = \begin{bmatrix} 1 & 0 & \frac{\sin(5\pi/6\Delta t)}{\Delta t} & \frac{1-\cos(5\pi/6\Delta t)}{\Delta t} \\ 0 & 1 & \frac{-(1-\cos(5\pi/6\Delta t))}{\Delta t} & \frac{\sin(5\pi/6\Delta t)}{\Delta t} \\ 0 & 0 & \frac{\cos(5\pi/6\Delta t)}{\Delta t} & \frac{\sin(5\pi/6\Delta t)}{\Delta t} \\ 0 & 0 & \frac{-\sin(5\pi/6\Delta t)}{\Delta t} & \frac{\cos(5\pi/6\Delta t)}{\Delta t} \end{bmatrix}$$

The process noise, $\boldsymbol{v}_{k,t}$, is assumed to be Gaussian distributed, with a zero mean and a covariance matrix given by Eq. (2.12). We chose $\epsilon_1 = \epsilon_2 = \epsilon_3 = 5 \times 10^{-1}$ in this chapter. The initial positions of the targets are set to be $(450, 300)$, $(480, 200)$ and $(400, 350)$, respectively, whereas the initial velocities are set to be $(0, -10)$, $(5, 0)$ and $(15, 0)$ respectively. The second and the third targets make one co-ordinated turn each during the tracking duration and the first target makes two co-ordinated turns. For each target the mode transition matrix $\boldsymbol{P}$ is given as

$$\boldsymbol{P} = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix}.$$

In Fig. (6.1), we plot the actual and the estimated trajectories using the standard particle filter (SPF) which implements the weight update following Eq. (6.14), the IMM-SSPF, the IMM-IPPF and the IMM-CPF for the three tracking scenarios. We used 1000 particles for the simulations and averaged the results over 25 Monte-Carlo iterations. We represented the error variance in the x− and the y− axes using ellipsoids with the axis length equal to the standard deviation. It can be seen that the estimated trajectories became less accurate as the difficulty of the problem increased. Further, the standard particle filter is consistently outperformed by all the other algorithms. This is due to the fact that the other algorithms all partition the state-space and operate over the individual partitions, and the performance of the filtering improved as the dimension of the state-space decreased. We also observed that the performance of the IMM-IPPF rapidly degraded as the problem difficulty increased. This proves that the PMHT assumption (A3) required to obtain the IMM-IPPF does not hold as the clutter density increases and the probability of detection decreases. Hence, this assumption has a harmful effect on the data association which degraded the performance. The IMM-CPF performed consistently well compared to the relative performance of other algorithms. The performance improvement is due to the fact that IMM-CPF does not use the PMHT assumption.

In Figs. (6.2), (6.3), (6.4), we plot the cumulative root mean squared-error (CRMSE) in the state estimates as a function of the tracking index for the easy, medium and hard scenarios, respectively. The CRMSE was computed as:

$$\text{CRMSE}_t = \frac{1}{4}\sum_{m=1}^{4}\sqrt{\frac{1}{N_{\text{mc}}K}\sum_{k=1}^{K}\sum_{i=1}^{N_{\text{mc}}}(\hat{x}_{k,t,i,m} - x_{k,t,m})^2}, \tag{6.50}$$

where $\hat{x}_{k,t,i,m}$ is the estimate of the m^{th} component of the k^{th} target state vector at time t in the i^{th} Monte-Carlo simulation. It can be seen that the performance of IPPF degraded rapidly as the problem complexity increased. DPF on the other hand performed well and its performance was similar to the performance obtained by SSPF.

In the next set of figures we compare performance of the algorithms as a function of the number of particles. In Fig. (6.5), we plot total CRMSE (TCRMSE) as a function of the number of the particles. The TCRMSE was computed as:

$$\text{TCRMSE} = \frac{1}{T}\sum_{t=1}^{T}\text{CRMSE}_t. \tag{6.51}$$

It can be seen that the performance of the filters improved as the number of particles increased. However, this performance improvement was not significant as the problem became harder.

In order to quantify the performance of data association and maneuver tracking, we defined two other performance metrics: incorrect to correct association ratio (ICAR) and the incorrect to correct mode ratio (ICMR) which are defined as follows

$$\text{ICAR} = \frac{1}{TN_{\text{mc}}}\sum_{i=1}^{N_{\text{mc}}}\sum_{t=1}^{T}\frac{\sum_{k=1}^{K} In(\lambda_{k,t,i} \neq \hat{\lambda}_{k,t,i})}{\sum_{k=1}^{K} In(\lambda_{k,t,i} = \hat{\lambda}_{k,t,i}) + \epsilon} \tag{6.52}$$

$$\text{ICMR} = \frac{1}{TN_{\text{mc}}}\sum_{i=1}^{N_{\text{mc}}}\sum_{t=1}^{T}\frac{\sum_{k=1}^{K} In(\alpha_{k,t,i} \neq \hat{\alpha}_{k,t,i})}{\sum_{k=1}^{K} In(\alpha_{k,t,i} = \hat{\alpha}_{k,t,i}) + \epsilon}, \tag{6.53}$$

where $In(\cdot)$ is an indicator function, and $\epsilon > 0$ is a small constant. In this work, we choose ϵ to be 1×10^{-1}. It can be seen from the definitions of ICAR and ICMR that the smaller these values, the better the performance of association and mode

tracking. We plot ICAR and ICMR as a function of number of the particles for various algorithms in Figs. (6.6) and (6.7), respectively.

It can be seen that for all the algorithms both ICAR and ICMR increased as the problem complexity increased. The ICAR of IMM-IPPF increased significantly whereas this increase was not significant for other algorithms when a larger number of particles were used. All the algorithms except SPF had a similar ICMR for an easy problem, but ICMR of IMM-IPPF increased as the problem complexity increased. ICMR of IMM-CPF was comparable to that of IMM-SPF for a hard problem. It can also be seen that the decrease in the ICAR and ICMR with the number of particles was not significant as the problem complexity increased.

In Table 6.2, we tabulate the average time taken (in seconds) for each of the algorithms for various problem scenarios and number of particles. IMM-SSPF took a longer time compared to any of the other methods. IMM-IPPF was the fastest when the number of particles was small, but IMM-CPF became slightly better as the number of particles increased. The computational time of all the algorithms increased slightly as the problem complexity increased.

In Fig. (6.8), we plot the empirical receiver operating characteristic (ROC) graphs obtained by employing IMM-CPF with 1000 particles for a hard problem for various values of μ_2, with μ_1 fixed. For this simulation, we choose $\mu_1 = 301$, since $d_{\max} = 150$ for the values of the measurement noise that we used for the experiments. It can be seen that for a given value of μ_2, as the probability of detection increased, the probability of false alarm also increased. It can also be seen that for a fixed false alarm probability, the probability of detection increases (or equivalently the probability of

		No of Particles				
		100	300	500	800	1000
Easy	SPF	0.08	0.25	0.42	0.67	0.85
	SSPF	0.25	0.76	1.25	2.02	2.58
	DPF	0.07	0.09	0.15	0.26	0.33
	IPPF	0.07	0.19	0.32	0.54	0.71
Medium	SPF	0.1	0.45	0.50	0.59	1.00
	SSPF	0.33	0.96	1.52	1.57	3.08
	DPF	0.12	0.44	0.36	0.69	0.53
	IPPF	0.08	0.30	0.41	0.52	0.84
Hard	SPF	0.12	0.36	0.56	0.91	1.09
	SSPF	0.36	1.07	1.70	2.76	3.33
	DPF	0.24	0.25	0.28	0.61	0.72
	IPPF	0.10	0.29	0.46	0.76	0.95

Table 6.2: Average execution time for various algorithms.

miss decreases) with the decrease in the value of μ_2. This behavior supports our discussion following Lemma 7.

In Fig. (6.9), we plot the weights given to each of the strategies by the regret matching algorithm at a given time, as a function of the iteration number. We used 1000 particles for a hard problem. It can be seen that the weight given to one strategy goes to one, while the other weights converge to zero. The same behavior is exhibited by all the targets. We obtained similar results for all the problem scenarios. The same behavior was seen as the number of particles changed for a given problem scenario. This simulation supports our observation empirical distributions of the regret matching always lead to a pure strategy equilibrium in data association games.

6.8 Summary

In this chapter, we developed Monte Carlo methods for tracking maneuvering targets in clutter. First, we extended the sequential sampling particle filter (SSPF) and the independent partition particle filter (IPPF) of [82] to the case of maneuvering targets, which we call the IMM-SSPF and the IMM-IPPF, respectively. In IMM-SSPF, the association samples the individual targets are drawn sequentially by factorizing the importance weights, and in IMM-IPPF, the samples are drawn independent of each other. The independence assumption leads to a faster implementation for IMM-IPPF, and provides a good performance when the clutter rate is low and the probability of detection is high. However, the performance degrades significantly at higher clutter rate and the probability of detection decreased.

With this motivation, we developed a new particle filtering technique which we call the interacting multi-model concurrent particle filtering (IMM-CPF) that uses game-theory to solve the data association. In IMM-CPF, we solve the association problem by formulating it as a game among the trackers. We defined the strategy spaces and the utility functions of each player, and we used regret matching to find the equilibrium of the game. We showed using numerical simulations that regret matching converges to a pure strategy equilibrium. We derived sufficient conditions for the resulting equilibrium to be a feasible data association vector, and necessary and sufficient conditions for the equilibrium to result in false alarms and missed detections.

We evaluated the performance of the algorithms by providing a number of simulation results. We considered synthetic tracking problems of different complexities for the simulations, and compared the performance of the algorithms. We observed that all the algorithms showed a superior performance compared to the standard particle

filtering (SPF). The performance of the IMM-IPPF degraded rapidly as the difficulty of the problem increased. For this scenario, the IMM-SSPF performed well at a cost of additional computational complexity. Our proposed algorithm, the IMM-CPF showed similar performance as IMM-SSPF as the problem complexity increased, but takes significantly less computational time. Further, the IMM-CPF can be implemented in a parallel fashion and this implementation provides an added advantage, especially when the number of targets becomes large. The IMM-CPF also had fewer association errors compared to IMM-SSPF.

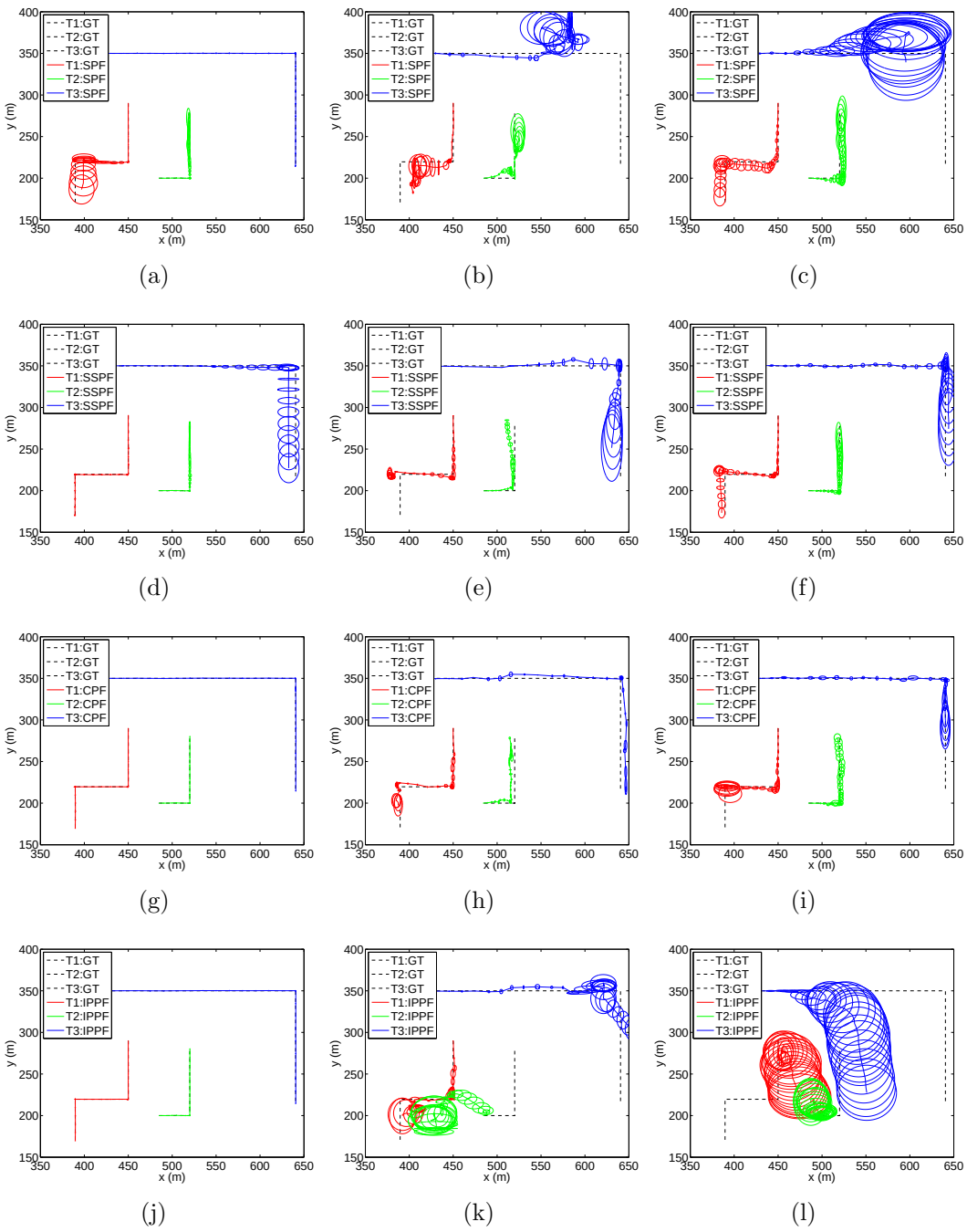


Figure 6.1: Estimated vs the actual trajectories of the targets obtained using different algorithms for (a), (d), (g), (j) easy, (b), (e), (h), (k) medium, and (c), (f), (i), (l) hard problems.

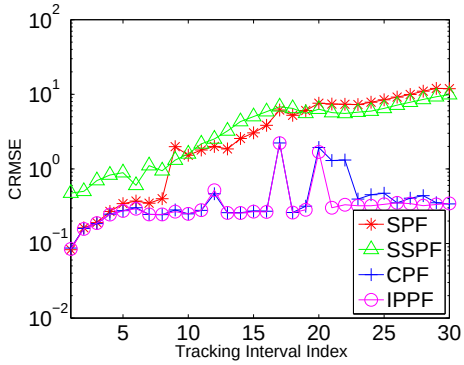


Figure 6.2: CMRSE in the state estimation of various algorithms for an easy problem.

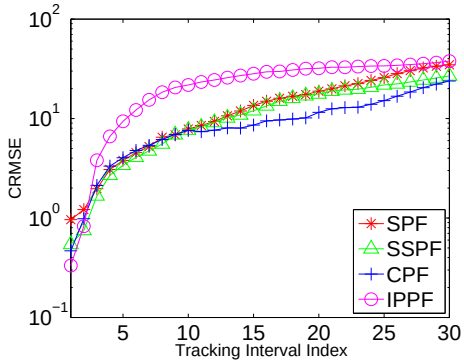


Figure 6.3: CMRSE in the state estimation of various algorithms for medium problem.

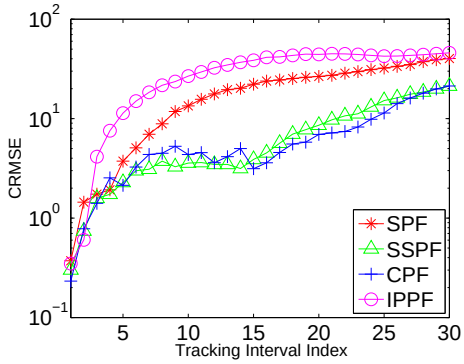
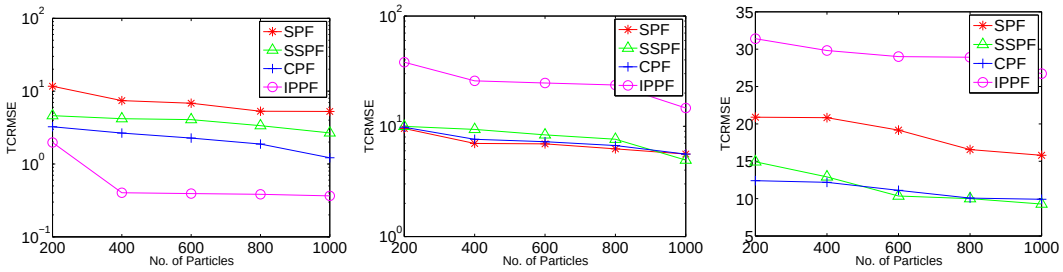
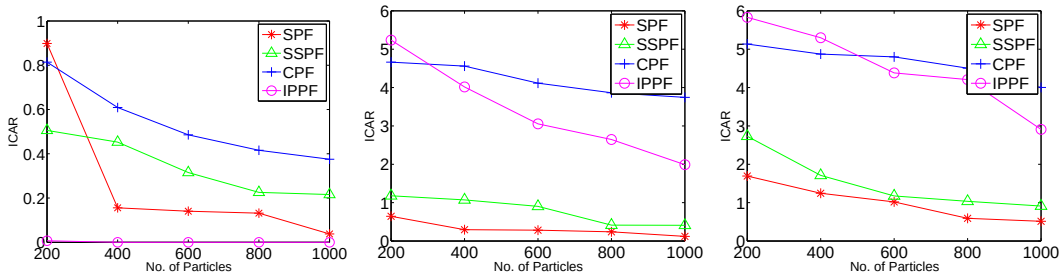


Figure 6.4: CMRSE in the state estimation of various algorithms for hard problem.



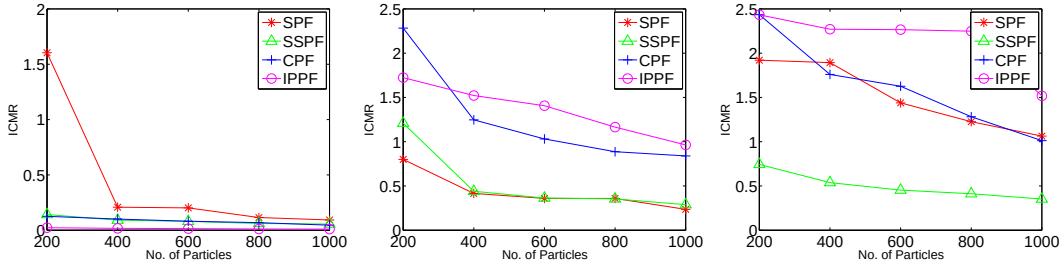
(a) (b) (c)

Figure 6.5: TCRMSE of various algorithms as a function of number of particles for (a) easy, (b) medium and (c) hard problems.



(a) (b) (c)

Figure 6.6: ICAR of various algorithms as a function of number of particles for (a) easy, (b) medium and (c) hard problems.



(a) (b) (c)

Figure 6.7: ICMR of various algorithms as a function of number of particles for (a) easy, (b) medium and (c) hard problems.

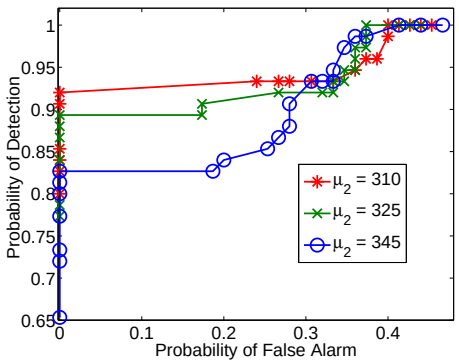
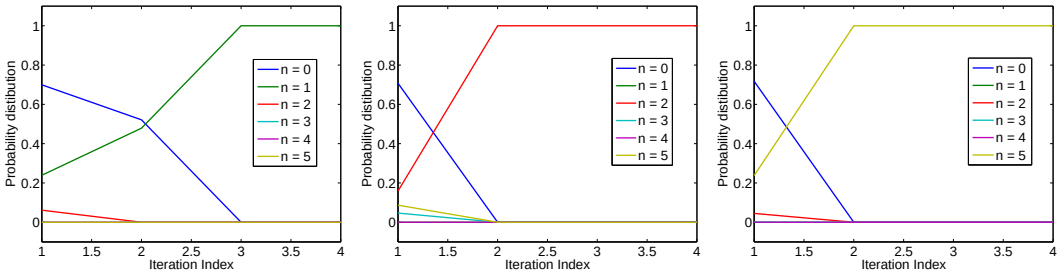


Figure 6.8: ROC graph for various values of μ_2.



(a) (b) (c)

Figure 6.9: Probability distributions over the strategies as a function of the iteration number for (a) Target 1 (b) Target 2 and (c) Target 3.

Chapter 7

Conclusions

In this dissertation, we developed computationally efficient tracking algorithms for multiple-target tracking in three complex scenarios: (i) targets moving in a time-varying multipath environment; (ii) unknown and time-varying number and types of targets; (iii) targets with low probability of detection moving in dense clutter. We will first summarize the contributions of this dissertation and then provide directions in which this work can be extended.

7.1 Summary of Contributions

We proposed a sparsity-based algorithm for tracking targets moving in a time-varying multipath environment. The channel state that describes the multipath environment was assumed to be known. We developed a sparse measurement model for the received signal, by considering a finite dimensional representation of the system function that characterizes the multipath environment and then reformulated the problem of MTT as a block support recovery problem from the sparse measurement model. We exploited the structure of the dictionary matrix to develop a computationally efficient

block support recovery algorithm (and thereby a multiple-target tracking algorithm). Further, we also derived an upper bound on the overall error probability of wrongly identifying the support of the sparse signal.

Next, we developed a particle filter for jointly estimating both the target and the channel states (when the channel state is unknown to the radar). The dimension of the overall state is high, and hence the standard particle filter requires a large number of particles making it computationally expensive. We exploited the structure in the measurement model to develop Multiple Rao-Blackwellized Particle Filter (MRBPF) to jointly estimate both the target and the channel state. MRBPF is a hybrid filter that uses one particle filter for each target's state partition, and uses a Kalman filter to analytically find an estimate of the channel state. Since MRBPF operates on low dimensional subspaces, it reduces the complexity involved with high-dimensional state space. We also computed the posterior Cramér-Rao bound (PCRB) on the estimates of the target state and the channel state. Using PCRB as a metric, we obtained a suitable subset of antennas to be used for transmission in each tracking interval, as well as the power to be transmitted by these antennas.

We then considered the problem of tracking an unknown and time-varying number and types of targets using a multi-modal sensor network. In a multi-modal sensor network, different quantities associated with the same state are measured using sensors of different kinds. We first developed a general filtering technique to estimate the unknown state from the multi-modal measurements for a special class of problems which can be modeled hierarchically. The proposed filter, the Hierarchical Particle Filter (HPF), estimates the global posterior density of the unknown state in multiple stages, by partitioning the state and the measurement spaces into lower dimensional

subspaces. At each stage, an estimate of one partition is obtained using the measurements from the corresponding partition, and the information from the previous stages. In this way, the proposed filter combines the information from different sensors. We then model the problem of MTT using a hierarchical model and demonstrate the proposed filtering for joint initiation, termination and tracking of an unknown number of targets.

We also proposed a unified framework for multi-modal sensor management that comprises sensor selection (SS), resource allocation (RA) and data fusion (DF). Our approach is inspired by the trading behavior of economic agents in commercial markets. We model each sensor node (SN) as a seller who wants to sell the data it collects, the sensor network manager (SM) as a buyer, and the interaction among them as a double sided market with both consumers and producers. We proposed an iterative double auction mechanism for computing the equilibrium of such a market. We relate the equilibrium point to the solutions of SS, RA and DF.

We also proposed a new particle filter, the interacting multi-model concurrent particle filter (IMM-CPF) to track multiple maneuvering targets in the presence of association ambiguities. We treated the data association and state estimation as separate subproblems and we developed a game-theoretic framework to address the data association. We modeled each tracker as a player and the set of measurements as strategies. We developed utility functions for each player, and then used a regret-based learning algorithm to find the equilibrium of this game. The game-theoretic approach allowed us to associate measurements to all the targets simultaneously.

7.2 Future Directions

There are several directions in which the work presented in this dissertation can be extended. For sparsity-based tracking, the problem of dictionary design is of great interest. Such a design will enable the sparsity-based recovery algorithms to become more robust to the grid mismatch problem. There has been a lot of interest in designing *optimal* dictionary matrices, however to the best of our knowledge all such designs are restricted to the static case, i.e., design based only on the current measurements. Since the problem of MTT is inherently dynamic, it it natural to think of a design paradigm that will use all the measurements available and the predicted value of the state to efficiently design a dictionary matrix for each time.

The problems of analyzing the convergence rate of the double auction mechanism and the regret matching algorithm proposed in this dissertation is an interesting mathematical problem and needs further research.

Recently, there has been a growing interest in using graphical model to address inference problems. Graphical models are successfully employed in a lot of communication and signal processing applications. However, most of these applications use linear models and hence the message updates can be computed analytically. For the problem of MTT, since the models are nonlinear we should resort to particle based approximations. Combining the particle methods with graphical models is a very interesting direction for future research. Several practical problems can be addressed within this framework if computationally feasible algorithms that use particle methods for message passing in dynamic graphical models can be developed.

We would also like to extend the proposed methods to deal with scenarios where the measurements at different sensors are correlated. Developing efficient particle filtering algorithms to exploit this correlation will be an interesting research problem.

Finally, validating the performance of our proposed techniques with real radar data is an important issue, and needs to be addressed in the future.

Appendix A

Proof of Theorem 4

The log-likelihood of the measurement vector can be written as

$$\begin{aligned}\log p(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{t+1}) &= \log \prod_{p\in\pi_{t+1}} p(\boldsymbol{y}_{p,t+1}|\boldsymbol{\xi}_{t+1}) \\ &= \sum_{p\in\pi_{t+1}} (\boldsymbol{y}_{p,t+1}-\boldsymbol{\mu}^p_{t+1})^H \boldsymbol{\Sigma}^{-1}_{\mathrm{w},p} (\boldsymbol{y}_{p,t+1}-\boldsymbol{\mu}^p_{t+1}), \end{aligned} \tag{A-1}$$

where

$$\begin{aligned}\boldsymbol{\mu}^p_{t+1} &= \sum_{q\in\pi_{t+1}} \sum_{k\in\{1,\cdots,K\}} \sqrt{\gamma_q}\boldsymbol{\Phi}_{pqk}\boldsymbol{\beta}_{pqk}, \\ &= \sum_{q\in\pi_{k+1}} \sqrt{\gamma_q}\boldsymbol{\mu}^{pq}. \end{aligned} \tag{A-2}$$

The Hessian of the log-likelihood with respect to the complex vector, $\boldsymbol{\xi}_{t+1}$, is evaluated as follows.

$$
\begin{aligned}
& -\Delta_{\xi_{t+1}}^{\xi_{t+1}} \log p(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{t+1}) \\
= \; & -\Delta_{\xi_{t+1}} \left\{ \Delta_{\xi_{t+1}} \log p(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{t+1}) \right\}^H \\
= \; & -\sum_{p \in \pi_{t+1}} \Delta_{\xi_{t+1}} \left\{ \Delta_{\xi_{t+1}} (\boldsymbol{y}_{p,t+1} - \boldsymbol{\mu}_{t+1}^p)^H \boldsymbol{\Sigma}_{\mathrm{w},p}^{-1} (\boldsymbol{y}_{p,t+1} - \boldsymbol{\mu}_{t+1}^p) \right\}^H \\
= \; & -\sum_{p \in \pi_{t+1}} \Delta_{\xi_{t+1}} \left\{ \left(\frac{\partial \boldsymbol{\mu}_{t+1}^p}{\partial \boldsymbol{\xi}_{t+1}} \right) \frac{\partial}{\partial \boldsymbol{\mu}_{t+1}^p} (\boldsymbol{y}_{p,t+1} - \boldsymbol{\mu}_{t+1}^p)^H \boldsymbol{\Sigma}_{\mathrm{w},p}^{-1} (\boldsymbol{y}_{p,t+1} - \boldsymbol{\mu}_{t+1}^p) \right\}^H \\
= \; & -2 \sum_{p \in \pi_{t+1}} \Re \left\{ \Delta_{\xi_{t+1}} \boldsymbol{\Sigma}_{\mathrm{w},p}^{-1} \left(\boldsymbol{y}_{p,t+1} - \boldsymbol{\mu}_{t+1}^p \right) \left(\frac{\partial \boldsymbol{\mu}_{t+1}^p}{\partial \boldsymbol{\xi}_{t+1}} \right)^H \right\} \\
= \; & -2 \sum_{p \in \pi_{t+1}} \Re \left\{ \boldsymbol{\Sigma}_{\mathrm{w},p}^{-1} \left(\boldsymbol{y}_{p,t+1} - \boldsymbol{\mu}_{t+1}^p \right) \frac{\partial}{\partial \boldsymbol{\xi}_{t+1}} \left(\frac{\partial \boldsymbol{\mu}_{t+1}^p}{\partial \boldsymbol{\xi}_{t+1}} \right)^H - \left(\frac{\partial \boldsymbol{\mu}_{t+1}^p}{\partial \boldsymbol{\xi}_{t+1}} \right) \boldsymbol{\Sigma}_{\mathrm{w},p}^{-1} \left(\frac{\partial \boldsymbol{\mu}_{t+1}^p}{\partial \boldsymbol{\xi}_{t+1}} \right)^H \right\}.
\end{aligned}
\tag{A-3}
$$

Therefore we have

$$
\mathbb{E}_{\boldsymbol{y}|\xi} \left[-\Delta_{\xi_{t+1}}^{\xi_{t+1}} \log p(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{t+1}) \right] = 2 \sum_{p \in \pi_{t+1}} \Re \left\{ \left(\frac{\partial \boldsymbol{\mu}_{t+1}^p}{\partial \boldsymbol{\xi}_{t+1}} \right) \boldsymbol{\Sigma}_{\mathrm{w},p}^{-1} \left(\frac{\partial \boldsymbol{\mu}_{t+1}^p}{\partial \boldsymbol{\xi}_{t+1}} \right)^H \right\}. \tag{A-4}
$$

Substituting the value of $\boldsymbol{\mu}_{t+1}^p$, we get

$$
\mathbb{E}_{\boldsymbol{y}|\xi} \left[-\Delta_{\xi_{t+1}}^{\xi_{t+1}} \log p(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{t+1}) \right] = 2 \sum_{p,q,r \in \pi_{t+1}} \sqrt{\gamma_q \gamma_r} \Re \left\{ \left(\frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \boldsymbol{\xi}_{t+1}} \right) \boldsymbol{\Sigma}_{\mathrm{w},p}^{-1} \left(\frac{\partial \boldsymbol{\mu}_{t+1}^{pr}}{\partial \boldsymbol{\xi}_{t+1}} \right)^H \right\}. \tag{A-5}
$$

Appendix B

Evaluation of Partial Derivatives

The partial derivative $\frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \boldsymbol{\xi}_{t+1}}$ can be computed as follows. First, the vector $\boldsymbol{\xi}_{t+1}$ is partitioned as

$$\boldsymbol{\xi}_{t+1} = \left[(\boldsymbol{\theta}_{t+1}^{1})^T, \cdots, (\boldsymbol{\theta}_{t+1}^{M})^T, (\Re\{\boldsymbol{\beta}_{t+1}^{11}\})^T, (\Im\{\boldsymbol{\beta}_{t+1}^{11}\})^T, \cdots, (\Re\{\boldsymbol{\beta}_{t+1}^{PP}\})^T, (\Im\{\boldsymbol{\beta}_{t+1}^{PP}\})^T\right]^T \tag{B-1}$$

Following the definition of the complex vector differentiation [143], we have

$$\begin{aligned}
\frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \boldsymbol{\xi}_{t+1}} &= \left[\frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \boldsymbol{\theta}_{t+1}^{1}}, \cdots, \frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \boldsymbol{\theta}_{t+1}^{M}}, \frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \boldsymbol{\beta}_{t+1}^{11}}, \cdots, \frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \boldsymbol{\beta}_{t+1}^{PP}}\right], \\
&= \left[\frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \boldsymbol{\theta}_{t+1}}, \frac{1}{2}\left(\frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \Re\{\boldsymbol{\beta}_{t+1}^{11}\}} - j\frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \Im\{\boldsymbol{\beta}_{t+1}^{11}\}}\right), \cdots, \frac{1}{2}\left(\frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \Re\{\boldsymbol{\beta}_{t+1}^{PP}\}} - j\frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \Im\{\boldsymbol{\beta}_{t+1}^{PP}\}}\right)\right],
\end{aligned} \tag{B-2}$$

where the partial derivatives with respect to the target state and the channel state vector can be derived as,

- $\frac{\partial \boldsymbol{\mu}_{t+1}^{pq}}{\partial \boldsymbol{\theta}_{t+1}^{k}} = \left(\boldsymbol{\beta}_{pqk}^T \otimes \boldsymbol{I}_{LN}\right)\frac{\partial \mathrm{vec}(\boldsymbol{\Phi}_{pqk})}{\partial \boldsymbol{\varphi}_{t+1}^{pqk}}\boldsymbol{\Lambda}^{pqk}$
- $\boldsymbol{\varphi}^{pqk} = [\tau_{pqk}, \nu_{pqk}]^T$ is a vector of the delay-Dopplers corresponding to the k^{th} target and the q^{th}, p^{th} transmit-receive pair

- $\mathbf{\Lambda}^{pqk} = \frac{\partial \boldsymbol{\varphi}^{pqk}}{\partial \boldsymbol{\theta}^{k}_{t+1}}$ is a 2×4 matrix
- $\frac{\partial \boldsymbol{\mu}^{pq}_{t+1}}{\partial \Re\{\boldsymbol{\beta}^{uv}_{t+1}\}} = \frac{\partial \boldsymbol{\mu}^{pq}_{t+1}}{\partial \Im\{\boldsymbol{\beta}^{uv}_{t+1}\}} = \delta(p-u)\delta(q-v)\mathbf{\Phi}_{pqk}$

Here, the matrix $\mathbf{\Lambda}^{pqk}$ is given as: $\mathbf{\Lambda}^{pqk} = \begin{bmatrix} \lambda^{pqk}_{11} & \lambda^{pqk}_{12} & \lambda^{pqk}_{13} & \lambda^{pqk}_{14} \\ \lambda^{pqk}_{21} & \lambda^{pqk}_{22} & \lambda^{pqk}_{23} & \lambda^{pqk}_{24} \end{bmatrix}$. The elements of $\mathbf{\Lambda}^{pqk}$ are given as [144]

$$
\begin{aligned}
\lambda^{pqk}_{11} = \frac{\partial \tau_{pqk}}{\partial x_m} &= \frac{1}{c}\left(\frac{x_q - x_k}{R_{qk}} + \frac{x_p - x_k}{R_{pk}}\right), \\
\lambda^{pqk}_{12} = \frac{\partial \tau_{pqk}}{\partial y_k} &= \frac{1}{c}\left(\frac{y_q - y_k}{R_{qk}} + \frac{y_p - y_k}{R_{pk}}\right), \\
\lambda^{pqk}_{13} = \frac{\partial \tau_{pqk}}{\partial \dot{x}_k} &= 0, \\
\lambda^{pqk}_{14} = \frac{\partial \tau_{pqk}}{\partial \dot{y}_k} &= 0, \\
\lambda^{pqk}_{21} = \frac{\partial \nu_{pqk}}{\partial x_k} &= \frac{f_c}{c}\left(\frac{\dot{x}_k}{R_{qk}} - \dot{R}_{qk}\frac{x_q - x_k}{R^2_{qk}} + \frac{\dot{x}_k}{R_{pk}} - \dot{R}_{pk}\frac{x_p - x_k}{R^2_{pk}}\right), \\
\lambda^{pqk}_{22} = \frac{\partial \nu_{pqk}}{\partial y_k} &= \frac{f_c}{c}\left(\frac{\dot{y}_k}{R_{qk}} - \dot{R}_{qk}\frac{y_q - y_k}{R^2_{qk}} + \frac{\dot{y}_k}{R_{pk}} - \dot{R}_{pk}\frac{y_p - y_k}{R^2_{pk}}\right), \\
\lambda^{pqk}_{23} = \frac{\partial \nu_{pqk}}{\partial \dot{x}_k} &= \frac{f_c}{c}\left(\frac{x_q - x_k}{R_{qk}} + \frac{x_p - x_k}{R_{pk}}\right), \\
\lambda^{pqk}_{24} = \frac{\partial \nu_{pqk}}{\partial \dot{y}_k} &= \frac{f_c}{c}\left(\frac{y_q - y_k}{R_{qk}} + \frac{y_p - y_k}{R_{pk}}\right).
\end{aligned}
\tag{B-3}
$$

References

[1] Y. Bar-Shalom, P. Willet, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*, 3rd ed. YBS Publishing, 2011.

[2] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: Wiley, 2001.

[3] X.-R. Li, Y. Bar-Shalom, and T. Kirubarajan, *Estimation, Tracking and Navigation: Theory, Algorithms and Software*. New-York: John Wiley and Sons, Jun. 2001.

[4] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House Noorwood, MA, 1999, vol. 685.

[5] A. García-Fernàndez, "Detection and tracking of multiple targets using wireless sensor networks," PhD thesis, Universidad Politécnica de Madrid, Madrid Spain, 2011.

[6] D. Lane, M. Chantler, and D. Dai, "Robust tracking of multiple objects in sector-scan sonar image sequences using optical flow motion estimation," *Oceanic Engineering, IEEE Journal of*, vol. 23, no. 1, pp. 31–46, 1998.

[7] I. Cox, J. Rehg, and S. Hingorani, "A bayesian multiple-hypothesis approach to edge grouping and contour segmentation," *International Journal of Computer Vision*, vol. 11, no. 1, pp. 5–24, 1993. [Online]. Available: http://dx.doi.org/10.1007/BF01420590

[8] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, Dec. 2006.

[9] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *Robotics Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99 –110, Jun. 2006.

[10] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," *Robotics Automation Magazine, IEEE*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[11] F. P. Hans and D. V. Dijk, *Non-linear time series models in empirical finance*. Cambridge University Press, 2000.

[12] B. G. EP, G. M. Jenkins, and G. C. Reinsel, *Time series analysis: forecasting and control*. Wiley, 2013.

[13] B. Hammarberg, C. Forster, and E. Torebjork, "Parameter estimation of human nerve c-fibers using matched filtering and multiple hypothesis tracking," *Biomedical Engineering, IEEE Transactions on*, vol. 49, no. 4, pp. 329–336, 2002.

[14] Z. Chen, "Bayesian filtering: From Kalman filters to particle filters, and beyond," Adaptive Syst. Lab., McMaster Univ., Techical Report, 2003.

[15] S. M. Kay, *Fundamentals of Statistical Signal Processing, Estimation Theory*. Prentice Hall, 1993, vol. 1.

[16] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. on Automatic Control*, vol. 45, no. 3, pp. 477–482, Mar. 2000.

[17] M. Athans, R. P. Wishner, and A. Bertolini, "Suboptimal state estimation for continuous time nonlinear systems from discrete noisy measurements," *IEEE Trans. on Automatic Control*, vol. 13, pp. 504–514, 1968.

[18] M. S. Arulampalam, S. Maskell, N. J. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[19] B. Ristic, S. Arulampalam, and S. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[20] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte-Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[21] A. Doucet, "On sequential Monte-Carlo methods for Bayesian filtering," Dept. of Engr., Univ. Cambridge, UK, Tech. Rep, 1998.

[22] M. Bolic, P. M. Djuric, and S. Hong, "New resampling algorithms for particle filters," in *Proc. of Intl. Conf on Acoustics Speech and Signal Processing*, ser. 2, vol. 2. Hong Kong: IEEE, Apr. 2003, pp. 589–592.

[23] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian Bayesian state estimation," in *Proc. of Radar and Signal Processing*, vol. 140, no. 2. IEEE, Apr. 1993, pp. 107–113.

[24] M. F. Bugallo and P. Djurić, "Complex systems and particle filtering," in *Fourty Second Asilomar Conf. on Signals, Systems and Computers*. Asilomar, CA: IEEE, Nov. 2008, pp. 1183–1187.

[25] J. L. Krolik, J. Farrell, and A. Steinhardt, "Exploiting multipath propagation for GMTI in urban environments," in *IEEE Conference on Radar*. Verona, Newyork: IEEE, Apr. 24-27, 2006, pp. 65–68.

[26] B. D. Rigling, "Urban RF multipath mitigation," in *IET radar, sonar and navigation*, vol. 2, Dec. 2008, pp. 419–425.

[27] B. Krach and R. Weigel, "Markovian channel modeling for multipath mitigation in navigation receivers," in *European Conf. Antennas and Propagation*, Mar. 2009, pp. 1441–1445.

[28] B. Chakraborty, Y. Li, J. J. Zhang, T. Trueblood, A. Papandreou-Suppappola, and D. Morrell, "Multipath exploitation with adaptive waveform design for target tracking in urban terrain," in *International Conference on Acoustics, Speech, and Signal Processing*. Dallas, Texas: IEEE, Mar. 14-19, 2010, pp. 3894–3897.

[29] Y. Jin, J. M. Moura, and N. O'Donoughue, "Time reversal in multiple-input multiple-output radar," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 1, pp. 210–225, Feb. 2010.

[30] M. A. Herman and T. Strohmer, "High-resolution radar via compressive sensing," *IEEE Trans. on Signal Processing*, vol. 57, no. 6, pp. 2275–2284, Jun. 2009.

[31] S. Sen and A. Nehorai, "Sparsity-based multi-target tracking using OFDM radar," *IEEE Trans. on Signal Proc.*, vol. 59, no. 4, pp. 1902–1906, Apr. 2011.

[32] S. Gogineni and A. Nehorai, "Target estimation using sparse modeling for distributed MIMO radar," *IEEE Trans. on Signal Proc.*, vol. 59, pp. 5315–5325, Nov. 2011.

[33] M. R. Bell, "Information theory and radar waveform design," *IEEE Trans. on Information Theory*, vol. 39, no. 5, pp. 1578–1597, Sep. 1993.

[34] P. Z. Peebles, *Radar Principles*. New York: Wiley, 1998.

[35] P. Bello, "Characterization of randomly time-variant linear channels," *IEEE Trans. on Communications*, vol. 11, no. 4, pp. 360–393, Dec. 1963.

[36] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2001.

[37] A. Sayeed and B. Aazhang, "Joint multipath-doppler diversity in mobile wireless communications," *IEEE Trans. on Communications*, vol. 47, no. 1, pp. 123–132, Jan. 1999.

[38] N. Levanon and E. Mozeson, *Radar Signals*. New York : Wiley, 2004.

[39] Y. Wang, X. Li, and Y. Wang, "Novel spread-spectrum radar waveform," *Proc. of SPIE, Radar Sensor technologies*, vol. 3066, pp. 186–193, Jun. 1997.

[40] Z. S. Dobrosavljevic and M. L. Dukic, "A method of spread spectrum radar polyphase code design by nonlinear programming," *European Trans. on Telecommunications*, vol. 7, no. 3, pp. 239–242, May 1996.

[41] Y. C. Eldar and P. Kuppinger, "Block sparse signals: Uncertainity relations and efficient recovery," *IEEE Trans. on Signal Processing*, vol. 58, no. 6, pp. 3042–3054, Jun. 2010.

[42] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. on Computing*, vol. 24, no. 2, pp. 227–234, 1995.

[43] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. on Sci. Comp.*, vol. 20, no. 1, pp. 33–61, 1998.

[44] E. Candés, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate information," *Commun. Pure and Applied Mathematics*, vol. 59, pp. 1207–1233, Aug. 2006.

[45] E. J. Candés and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," *Annals of Statistics*, vol. 35, no. 6, pp. 2313–2351, 2007.

[46] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. on Information Theory*, vol. 50, pp. 2231–2242, Oct. 2004.

[47] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. on Information Theory*, vol. 53, pp. 4655–4666, Dec. 2007.

[48] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. and Comp. Harmonic Analysis*, vol. 26, pp. 301–321, May 2009.

[49] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. on Signal Processing*, vol. 56, pp. 2346–2356, Jun. 2008.

[50] D. P. Wipf and B. Rao, "Sparse bayesian learning for basis selection," *IEEE Trans. on Signal Processing*, vol. 52, pp. 2153–2164, Aug. 2004.

[51] G. Tang and A. Nehorai, "Performance analysis for sparse support recovery," *IEEE Trans. on Information Theory*, vol. 56, pp. 1383–1399, Mar. 2010.

[52] H. L. V. Trees, *Detection, Estimation and Modulation Theory.* New York: Wiley, 2001, vol. One.

[53] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, 1st ed. London, U.K: Wiley, 1965.

[54] R. A. Altes, "Target position estimation in radar and sonar, and generalized ambiguity analysis for maximum likelihood parameter estimation," *Proc. of IEEE*, vol. 67, no. 6, pp. 920–930, Jun. 1979.

[55] L. Xu, J. Li, and P. Stoica, "Target detection and parameter estimation for MIMO radar systems," *IEEE Trans. on Aerospace and Electronic Sys.*, vol. 44, no. 3, pp. 927–939, Jul. 2008.

[56] J. Min, R. Niu, and R. S. Blum, "Bayesian target location and velocity estimation for MIMO radar," *IET Sonar and Navigation*, no. 60972152, pp. 1–10, 2010.

[57] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming," Stanford University, http://stanford.edu/boyd/cvx, web page and software, Jun. 2009.

[58] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. on Signal Processing*, vol. 57, no. 2, pp. 451–462, Feb. 2009.

[59] H. Godrich, A. Petropulu, and H. V. Poor, "Resource allocation schemes for target localization in distributed multiple radar architectures," in *European Signal Processing Conf.* Aalborg, Denmark: IEEE, Aug., 23-27 2010.

[60] ——, "A combinatorial optimization framework for subset selection in distributed multiple-radar architectures," in *Intl. Conf. on Acoustics, Speech and Signal Processing.* Prague, Czech Republic: IEEE, May 2011.

[61] R. Tharmarasa, T. Kirubarajan, M. L. Hernandez, and A. Sinha, "PCRLB-based multisensor array management for multitarget tracking," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 43, no. 2, pp. 539–555, Apr. 2007.

[62] M. F. Bugallo, L. Ting, and P. M. Djuric, "Target tracking by multiple particle filtering," in *Aerospace Conf.* Big Sky, MT: IEEE, Mar. 2007, pp. 1–7.

[63] T. Schon, G. Gustafsson, and P. J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Trans. on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, Jul. 2005.

[64] M. Frederic, M. Bolic, and M. Bouchard, "Rao-Blackwellised particle filters: Examples of applications," in *Canadian Conf. on Electrical and Computer Engg.*, Ottawa, Ont., May 2006, pp. 1196–1200.

[65] A. Doucet, N. De Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*. Citeseer, 2000, pp. 176–183.

[66] P. Tichavsk, C. H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. on Signal Processing*, vol. 46, no. 5, May 1998.

[67] B. Han, S. W. Joo, and L. S. Davis, "Multi-camera tracking with adaptive resource allocation," *Intl. Jour. of Computer Vision*, vol. 91, pp. 45–58, Jan. 2011.

[68] E. Kenneth, A. Rajendra, N. Kannathal, and C. M. Lim, "Data fusion of multimodal cardiovascular signals," in *27th Annual Intl. Conf. of the Engineering in Medicine and Biology Society*. Shangai, China: IEEE, 2005, pp. 4689–4682.

[69] D. Chetverikov and Z. Jankó, "3d reconstruction by multimodal data fusion," *ERCIM News*, vol. 2010, no. 80, 2010.

[70] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proc. of the 13th Intl. Conf. on Very large data bases - Volume 30*, ser. VLDB '04. VLDB Endowment, 2004, pp. 588–599.

[71] P. K. Varshney, "Multisensor data fusion," *Elec. and Comm. Engg. Journal*, vol. 9, no. 6, pp. 245–253, December 1997.

[72] O. Punska, "Bayesian approaches to multi-sensor data fusion," M.Phil Thesis, University of Cambridge, Aug. 1999.

[73] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proc. of IEEE*, vol. 85, no. 1, pp. 6–23, January 1997.

[74] S. Bottone and C. Stanek, "Probabilistic graphical models and their application in data fusion," *Proc. of SPIE*, vol. 6566, no. 6566, 2007.

[75] M. Stone, "The opinion pool," *Annals of Statistics*, vol. 32, pp. 1339–1342, 1961.

[76] J. Manyika and H. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*. Ellis Horwood, New-York, 1994.

[77] B. Han, S. Joo, and L. Davis, "Probabilistic fusion tracking using mixture kernel-based bayesian filtering," in *Intl. Conf. on Computer Vision (ICCV)*, Oct. 2007, pp. 1–8.

[78] J. Vermaak, A. Doucet, and P. Perez, "Maintaining multimodality through mixture tracking," in *9th Intl. Conf. on Computer Vision*, vol. 2, Nice, France, 2003, pp. 1110–1116.

[79] B. Han, Y. Zhu, D. Comaniciu, and L. S. Davis, "Visual tracking by continuous density propagation in sequential Bayesian filtering framework," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 919–930, May 2009.

[80] D. Snyder, A. Hammoud, and R. White, "Image recovery from data acquired with a charge-coupled-device camera," *Jour. of the Optical Society of America*, vol. 10, no. 5, pp. 1014–1023, May 1993.

[81] M. Smith, "Bayesian sensor fusion: A framework for using multi-modal sensors to estimate target locations and identities in a battlefield scene," Ph.D. thesis, Department of Statistics, Florida State University, 2003.

[82] J. Vermaak, S. Godsill, and P. Perez, "Monte-Carlo filtering for multi-target tracking and data association," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 41, no. 1, pp. 309 – 332, Jan. 2005.

[83] P. Chavali and A. Nehorai, "Concurrent particle filtering and data association using game theory for tracking multiple maneuvering targets," *to appear in IEEE Trans. of Signal Processing*, 2013.

[84] H. Rowaihy, S. Eswaran, M. Johnson, D. Verma, A. Bar-Noy, T. Brown, and T. L. Porta, "A survey of sensor selection schemes in wireless sensor networks," in *Proc. of SPIE*, vol. 6562, no. 1, 2007, pp. A1–A13.

[85] D. L. Hall and S. A. McMullen, *Mathematical techiniques in Multisensor Data Fusion*, 2nd ed. Boston: Artech House, 2004.

[86] T. A. Weber, *Oxford Handbook on Pricing Management*. Oxford University Press, to appear in 2010.

[87] T. Mullen, V. Avasarala, and D. L. Hall, "Customer-driven sensor management," *IEEE Intelligent Systems*, vol. 21, pp. 41–49, Mar. 2006.

[88] S. T. Shivappa, B. D. Rao, and M. Trivedi, "An iterative decoding algorithm for fusion of multimodal information," *EURASIP Jour. on Advances in Signal Processing*, vol. 2008, 2008.

[89] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *Proc. of Third Intl. symposium on Information Processing in Sensor Networks*, Apr. 2004, pp. 36–45.

[90] J. Lu, L. Bao, and T. Suda, "Coverage-aware sensor engagement in dense sensor networks," in *Proc. of Intl. Conf. on Embedded and Ubiquitous Computing*, Dec. 2005.

[91] M. Perillo and W. Heinzelman, "Optimal sensor management under energy and reliability constraints," in *Proc. of IEEE Conf. on Wireless Communications and Networking*. IEEE, Mar. 2003.

[92] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, no. 19, pp. 61–72, Mar. 2002.

[93] H. N. Psaraftis and A. N. Perakis, "A basic problem of resource allocation in target tracking," *Jour. of the Acoust. Soc of America*, vol. 72, no. 3, pp. 824–833, 1982.

[94] V. Farias and B. Vanroy, "Approximation algorithms for dynamic resource allocation," *Operations Research Letters*, vol. 34, no. 2, pp. 180–190, 2006.

[95] L. Walras, *Elements of pure Economics, or the theory of social wealth*, english ed. ed. Allen and Unwin, 1954.

[96] K. Arrow and G. Debreu, "Existence of an equilibrium for competitive economy," *Econometrica*, vol. 22, pp. 265–290, 1954.

[97] H. Scarf, "The computation of equilibrium prices," in *Applied General Equilibrium Analysis*, J. B. Shoven and H. Scarf, Eds. Cambridge: Cambridge University Press, 1984, pp. 1–49.

[98] A. Nagurney, *Network Economics: A Variantional Inequlity Approach*. Kluwer Academic Publishers, 1993.

[99] X. Deng, , C. Papadimitriou, and S. Safra, "On the complexity of equilibria," in *Proc. of ACM symposium on Theory of computing*. ACM, 2002, pp. 67–71. [Online]. Available: http://doi.acm.org/10.1145/509907.509920

[100] N. Devanur, C. Papadimitriou, A. Saberi, and V. Vazirani, "Market equilibrium via a primal-dual-type algorithm," in *Proc. of Foundations of Comp. Sci.*, Nov. 2002, pp. 389–395.

[101] N. Devanur and V. Vazirani, "An improved approximation scheme for computing Arrow-Debreu prices for the linear case," in *Proc. of Foundations of Software Tech. and Theoritical Comp. Sci.*, 2003, pp. 149–155.

[102] K. Jain, M. Mahdian, and A. Saberi, "Approximating market equilibrium," in *Workshop on Approximation algorithms for combinatorial optimization*, 2003.

[103] K. Jain, "A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities," in *Proc. of the 45th Annual IEEE Symposium on Foundations of Computer Science*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 286–294. [Online]. Available: http://portal.acm.org/citation.cfm?id=1032645.1033184

[104] D. Bertsekas, "The auction algorithm: A distributed relaxation method for the assignment problem," *Annals of Operations Research*, vol. 14, pp. 105–123, 1988.

[105] S. Parsons, J. A. Rodriguez-Aguilar, and M. Klein, "Auctions and bidding: A guide for computer scientists," *ACM Comput. Surveys*, vol. 43, no. 10, 2009. [Online]. Available: http://doi.acm.org/10.1145/1883612.1883617

[106] R. Garg and S. Kapoor, "Auction algorithms for market equilibrium," in *Proc. of the thirty-sixth annual ACM symposium on Theory of computing*. ACM, 2004, pp. 511–518. [Online]. Available: http://doi.acm.org/10.1145/1007352.1007430

[107] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*, 1st ed. Cambridge University Press, 2009. [Online]. Available: http://www.masfoundations.org

[108] D. Friedman and J. Rust, Eds., *The Double Auction Market: Institutions, Theories, and Evidence*. Addison Wesley, 1993.

[109] P. Cramton, Y. Shoham, and R. Steinberg, Eds., *Combinatorial Auctions*. Cambridge, MA, USA: MIT Press, 2006.

[110] L. Li, Y. Liu, D. Hausheer, and B. Stiller, "Design and evaluation of combinatorial double auction for resource allocation in grids," in *Proc. of sixth Intl. workshop on Internet charging and QoS technologies*. Aachen, Germany: Springer, May 2009.

[111] M. Xia, J. Stallaert, and A. B. Whinston, "Solving the combinatorial double auction problem," *European Jour. of Operational Research*, vol. 164, pp. 239–251, 2005.

[112] E. L. Lawler and D. E.Wood, "Branch-and-bound methods: A survey," *Oper. Res.*, vol. 14, pp. 699–719, 1966.

[113] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley:New York, 2006.

[114] A. Pandharipande, "Principles of OFDM," *IEEE Potentials*, vol. 21, no. 2, pp. 16–19, Apr. 2002.

[115] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J.Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, 1998.

[116] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Autom. Control*, vol. 33, pp. 780–783, 1988.

[117] S. McGinnity and G. Irwin, "Multiple model bootstrap filter for manoeuvring target tracking," *IEEE Trans. Aerospace and Electronic Systems*, vol. 36, no. 3, pp. 1006–1012, Jul. 2000.

[118] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. on Automatic Control*, vol. AC-24, pp. 843–854, 1979.

[119] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451 – 460, 1975.

[120] T. Fortman, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Jour. of Oceanic Engineering*, vol. 8, pp. 173–184, 1983.

[121] H. Gauvrit, J.-P. L. Cadre, and C. Jauffret, "A formulation of multitarget tracking as an incomplete data problem," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, pp. 1242–1257, Oct. 1997.

[122] N. Gordon, "A hybrid bootstrap filter for target tracking in clutter," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 353–358, Jan. 1997.

[123] C. Hue, J.-P. Le Cadre, and P. Perez, "Sequential Monte-Carlo methods for multiple target tracking and data fusion," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 309 –325, Feb. 2002.

[124] R. Karlsson and F. Gustafsson, "Monte-Carlo data association for multiple target tracking." *Proc. of the IEE International Seminar on Target Tracking:Algorithms and Applications*, vol. 1, pp. 1–5, Oct. 2001.

[125] S. Särkkä, A. Vehtari, and J. Lampinen, "Rao-Blackwellized Monte-Carlo data association for multiple target tracking," in *Proc. of the Seventh International Conference on Information Fusion*, Stockholm, Sweden, Jun. 2004, pp. 583–590.

[126] O. Hlinka, F. Hlawatsch, and P. Djuric, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 61–81, 2013.

[127] J. Beaudeau, M. Bugallo, and P. Djuric, "Target tracking with asynchronous measurements by a network of distributed mobile agents," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 3857–3860.

[128] F. Meyer, E. Riegler, O. Hlinka, and F. Hlawatsch, "Simultaneous distributed sensor self-localization and target tracking using belief propagation and likelihood consensus," in *Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2012, pp. 1212–1216.

[129] O. Simeone, U. Spagnolini, Y. Bar-ness, and S. H. Strogatz, "Distributed synchronization in wireless networks," *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 81–97, 2008.

[130] H. Gauvrit, J.-P. L. Cadre, and C. Jauffret, "A formulation of multitarget tracking as an incomplete data problem," *IEEE Trans. Aerosp. Electron. Syst.,*, vol. 33, p. 12421257, Oct. 1997.

[131] R. L. Streit and T. E. Luginbuhl, "Maximum likelihood method for probabilistic multi-hypothesis tracking," *Proc. SPIE*, vol. 2235, Apr. 1994.

[132] P.Willett, Y. Ruan, and R. Streit, "The PMHT for maneuvering targets," *Proc. SPIE*, vol. 3373, Apr. 1998.

[133] J. Liu and R. Chen, "Sequential Monte-Carlo methods for dynamical systems," *Jour. of American Statistical Assoc.*, vol. 93, no. 443, pp. 1032–1044, 1993.

[134] N. Ikoma and S. J. Godsill, "Extended object tracking with unknown association, missing observations, and clutter using particle filters," in *Proc. of 2003 IEEE Workshop on Statistical Signal Processing*. St. Louis, MO: IEEE, Sep. 2003, pp. 485–488.

[135] D. Fudenberg and J. Tirole, *Game theory*. Cambridge, Massachusetts: The MIT Press, 1991.

[136] D. Blackwell and M. A. Girshick, *Theory of games and statistical decisions*. New York, NY: John Wiley & Sons, Inc., 1954.

[137] C. H. Papadimitriou, *Algorithmic Game Theory*. New York, NY: Cambridge University Press, 2007, ch. The complexity of finding Nash equilibria, pp. 29–50.

[138] ——, "On the complexity of the parity argument and other inefficient proofs of existence," *Jour. of Computer and System Sciences*, vol. 48, no. 3, pp. 498–532, 1994.

[139] R. Aumann, "Subjectivity and correlation in randomized strategies," *Jour. of Mathematical Economics*, vol. 1, no. 1, pp. 67–96, 1974.

[140] C. H. Papadimitriou and T. Roughgarden, "Computing correlated equilibria in multi-player games," *Jour. of the ACM*, vol. 55, no. 3, pp. 1–29, Aug. 2008.

[141] S. Hart and D. Schmeidler, "Existence of correlated equilibria," *Mathematics of Operations Research*, vol. 14, pp. 18–25, 1989.

[142] S. Hart and A. Mas-Colell, "A simple adaptive procedure leading to correlated equilibrium," *Econometrica*, vol. 68, no. 5, pp. 1127–1150, 2000.

[143] A. Hjorungnes and D. Gesbert, "Complex-valued matrix differentiation: Techniques and key results," *IEEE Trans. on Signal Processing*, vol. 55, no. 6, pp. 2740–2746, Jun. 2007.

[144] J. Zhang, B. Manjunath, A. Papandreou-Suppappola, and D. Morell, "Dynamic waveform design for target tracking using MIMO radar," in *Asilomar Conference on Signals, Systems and Computers*. Asilomar, CA: IEEE, Oct. 2008, pp. 31–35.

Vita

Srinivas Phani Kumar Chavali

Degrees

Ph.D., Electrical & Systems Engineering, Washington University in St. Louis, MO, USA, October 2013
M.S., Electrical & Systems Engineering, Washington University in St. Louis, MO, USA, August 2010
B.Tech., Electronics & Communication Engineering, International Institute of Information Technology, Hyderabad, India, June 2007

Affiliations

Student member of IEEE Signal Processing Society

Awards

Finalist of the student paper competition at the *IEEE Sensor Array and Multichannel Processing Conference (SAM)*, Jun. 2012 and the *IEEE Computational Advances in Multi-Sensor Adaptive Processing Workshop*, Dec. 2011.

Publications

P. Chavali and A. Nehorai, "Scheduling and Resource Allocation in a Cognitive Radar Network for Multiple-Target Tracking", *IEEE Trans. on Signal Processing*, vol. 60, no. 2, pp. 715-729, Feb. 2012

P. Chavali and A. Nehorai, "A Low-Complexity Multi-Target Tracking Algorithm for Urban Environments using Sparse Modeling", *Signal Processing*, vol. 92, pp. 2199-2213, Sep. 2012

P. Chavali and A. Nehorai, "Managing Multi-Modal Sensor Networks using Price Theory", *IEEE Trans. on Signal Processing*, vol. 60, no. 9, pp. 4874-4887, Sep. 2012

P. Yang, P. Chavali, E. Gilboa, and A. Nehorai, "Parallel load schedule optimization with renewable distributed generators in smart grids," *in IEEE Trans. on Smart Grid*, vol. 4, no. 3, pp. 1431-1441, Sep. 2013.

P. Chavali and A. Nehorai, "Concurrent particle filtering and data association using game theory for tracking multiple maneuvering targets", *in IEEE Trans. on Signal Processing*, vol. 61, pp. 4934-4948, Oct. 2013.

P. Chavali and A. Nehorai, "Hierarchical Particle Filtering for Multi-Modal Sensor Networks With Application to Multiple-Target Tracking", to appear in *Signal Processing*.

P. Chavali, P. Yang, and A. Nehorai, "Distributed Appliance Scheduling in Smart Grid", in review for *IEEE Trans. on Smart Grid*.

P. Chavali and A. Nehorai, "Cognitive Radar for Target Tracking in Multipath Scenarios", in *Proc. of Intl. Conf. on Waveform Diversity and Design (WDD)*,pp. 110-114, Aug. 2010, Niagara Falls, Canada.

P. Chavali and A. Nehorai, "A Low Complexity Sparsity Based Multi-Target Tracking Algorithm for Urban Environments," in *Proc. of Radar Conf.*, pp. 309-315, May 2011, Kansas City, USA.

P. Chavali and A. Nehorai, "Multiple Rao-Blackwellized Particle Filtering for Target Tracking in Urban Environments", in *Proc. of Computational Advances in Multi-sensor Adaptive Processing (CAMSAP)*, pp. 409-412, Dec. 2011, San Juan, USA (student paper award finalist).

P. Chavali and A. Nehorai, "Price Theory Framework for Multiple Target Tracking", in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, pp. 5193-5196, Mar. 2012, Kyoto, Japan.

P. Chavali and A. Nehorai, "Hierarchical Particle Filtering in Multi-Modal Sensor Networks", in *Proc. of Seventh IEEE Sensor Array and Multichannel Signal Processing (SAM) Workshop*, pp. 149-152, Jun. 2012, Hoboken, NJ (student paper award finalist).

P. Yang, P. Chavali and A. Nehorai, "Parallel autonomous optimization of demand response with renewable distributed

generators", in *Proc. Smart Grid Communications*, pp. 55-60, Nov. 2012, Tainan city, Taiwan.

P. Chavali and A. Nehorai, "Distributed Data Association via Game Theory", in *Proc. of Radar Conf.*, May 2013, Ottawa, Canada.

E. Gilboa, P. Chavali, P. Yang and A. Nehorai, "Distributed Optimization via Adaptive Regularization for Large Problems with Separable Constraints", in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, May 2013, Vancouver, Canada.

December 2013