# Stony Brook University

**(M)ulti-(A)gent (S)ystems for (CO)operative (T)racking**

A Dissertation Presented

by

Jonathan Beaudeau

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

**Doctor of Philosophy**

in

**Electrical Engineering**

Stony Brook University

**December 2014**

**Stonybrook University**
The Graduate School

**Jonathan Beaudeau**

We, the dissertation commitee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

---

**Petar Djurić - Dissertation Adivsor**
**Professor, Electrical Engineering**

---

**Mónica Bugallo - Chairperson of Defense**
**Associate Professor, Electrical Engineering**

---

**John Murray - Committee Member**
**Associate Professor, Electrical Engineering**

---

**Eugene Feinberg - Committee Member**
**Distinguished Professor, Applied Mathematics and Statistics**

This dissertation is accepted by the Graduate School

---

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

**Multi-Agent Systems for Cooperative Tracking**

by

**Jonathan Beaudeau**

**Doctor of Philosophy**

in

**Electrical Engineering**

Stony Brook University

2014

This dissertation is a culmination of research in several parallel, yet related avenues dealing with statistical estimation in the context of dynamic state tracking. The main contribution of the thesis is to provide a novel scalable framework for tracking of a high-dimensional target state using a distributed and cooperative network of agents. This framework is incorporated into a specific application consisting of a multiple target tracking (MTT) environment, with each agent employing a number of mobile received-signal-strength (RSS) sensors capable of collecting measurements revealing localized information regarding the target state environment. The dimensionality of this problem, which is the dominating factor determining feasibility of any proposed solution, is effectively managed by dynamically assigning each agent a partition of the full target space to track. Agents are thus able to individually focus on a small piece of the full estimation problem and rely on inter-agent communication to compensate for state subspaces outside their estimation scope which may adversely influence their own measurements. The framework mentioned is coined Multi-Agent Systems for Cooperative Tracking or MASCOT.

Specific implementation details and associated challenges relating to this framework and the specific application considered are presented. Namely, inter-agent communication and the required information fusion necessary for RSS sensors is extensively investigated. Additionally, optimal RSS sensor placement within the described multi-target environment is addressed, greatly facilitating enhanced performance of the main subspace partitioning algorithm. The MASCOT framework has been fully implemented in the MATLAB modeling language; computer simulation results are presented, demonstrating algorithm performance and its superiority to earlier methods of handling such a problem. Finally, specific analysis of performance measures dealing with general statistical estimation (Bayesian estimation particularly) are presented as well as some novel results regarding a connection linking Frequentist/Bayesian estimation paradigms.

I dedicate this work to my wife Sofia Jürgensen, who has filled my life with absolute happiness and fulfillment since the day I met her. She has allowed me to recognize, appreciate, and truly savor the important things in life.

I also dedicate this work to my sister Stephanie Beaudeau, who has always treated her brother with love, kindness, and generosity. Her bravery, resilience, and compassion has been a true source of inspiration throughout my life.

Finally, I dedicate this work to my parents, Alain and Chantal Beaudeau and give them my *deepest gratitude*. They have been by my side supporting me in any way they can for all my life and are truly admirable in their unwavering commitment to see the happiness and success of their children. I would likely not be half of what I am today if it were not for them.

# Contents

**List of Figures** **vi**

**List of Tables** **x**

**1 Introduction** **1**

1.1 Main Problem Description . . . . . 1

1.1.1 MASCOT . . . . . 2

1.2 Detail-Oriented Challenges . . . . . 3

1.2.1 Measurement Interference . . . . . 3

1.2.2 Asynchronous Sensor Measurements . . . . . 4

1.3 Performance Metrics and Underlying Bounds . . . . . 5

**2 Dynamic State Estimation with MASCOT** **7**

2.1 Overview . . . . . 7

2.2 The general idea of MASCOT . . . . . 9

2.3 Introduction to the Multi-Target-Tracking Application . . . . . 12

2.4 Basic Target Tracking Background . . . . . 14

2.5 Cooperative-Agent Systems for Multi-Target Tracking . . . . . 17

2.6 Detailed MASCOT Implementation in the MTT Environment . . . . . 20

2.7 Target Tracking with a Mobile Sensor Network . . . . . 35

2.7.1 Positioning for Single-Target Estimation with Interference . . . . . 39

2.7.2 Positioning for Joint Estimation of Two Targets . . . . . 48

2.8 Performance Results . . . . . 56

2.8.1 Agent Cooperation . . . . . 56

2.8.2 Full MASCOT Implementation for Two Targets . . . . . 63

2.9 Concluding Remarks . . . . . 66

**3 Target Tracking in the Presence of Interference** **70**
3.1 Overview . . . 70
3.2 Interference Compensation through Source Localization . . . 71
3.3 Compensation by Dynamic Bias Tracking . . . 73
3.4 Performance Results . . . 75
3.5 Concluding Remarks . . . 78

**4 Multi-Mobile-Sensor Target Tracking with Asynchronous Measurements** **80**
4.1 Overview . . . 80
4.2 Proposed Solutions . . . 81
4.3 Performance Results . . . 86
4.4 Concluding Remarks . . . 89

**5 Bayesian Performance Metrics** **91**
5.1 Overview . . . 91
5.2 Introduction to the Bayesian-Bias Connection . . . 93
5.2.1 Related Work: The Optimal Bias Function . . . 93
5.2.2 Generalization of the Optimal Bias Function . . . 95
5.3 The Bias Connection in Estimation Problems of a Single Random Parameter . . . 97
5.3.1 Theoretical Results . . . 97
5.3.2 Single Parameter Estimation Performance Analysis with the Bias Connection 102
5.4 Extension to Sequential Estimation . . . 110
5.4.1 Overview . . . 110
5.4.2 Notation and Preliminary Definitions . . . 111
5.4.3 Vector Parameter Estimation . . . 113
5.4.4 Sequential Estimation . . . 114
5.4.5 Application to a Maneuvering Target Tracking Problem . . . 120
5.5 Concluding Remarks . . . 127

**6 Investigation of RSS-Type Multi-Target Fusion Likelihood Function** **129**
6.0.1 Single Sensor Theoretical Results . . . 130
6.0.2 Solution Method 1 . . . 131
6.0.3 Solution Method 2 . . . 132
6.1 Single Sensor Numerical Results . . . 135
6.1.1 Large Estimate Variances . . . 135
6.1.2 Small Estimate Variances . . . 139
6.1.3 Analytical Approximations For A Single Sensor . . . 140

6.1.3.1 Diagonal Covariance . . . . . 140
6.1.3.2 General Covariance Matrix . . . . . 143
6.2 Multiple Sensors . . . . . 149

**7 Conclusion and Future Work** **160**

**References** **162**

# List of Figures


2.1 MASCOT Agent Cooperation . . . . . 20

2.2 Plot of $\log \mathbb{B}_t(\Theta_t)$ for $(\sigma_v, \sigma_f) = (0.01, 0.04)$ for various values of $\Phi$ and $r = 2$. Blue lines indicate $\Theta_t^{(1)}$, and red indicate $\Theta_t^{(2)}$. . . . . 34

2.3 Plot of the ratio $\frac{\mathbb{B}_i\left(\Theta_t^{(1)}\right)}{\mathbb{B}_i\left(\Theta_t^{(2)}\right)}$ for $(\sigma_v, \sigma_f) = (0.01, 0.04)$ for various values of $\Phi$ and $r = 2$. 34

2.4 Plot of the ratio $\frac{\mathbb{B}_t\left(\Theta_t^{(1)}\right)}{\mathbb{B}_t\left(\Theta_t^{(2)}\right)}$ for $d = 2, \Phi = 16$ for various values of $\sigma_f$ and $r$. . . . . 35

2.5 Comparison of proposed solutions for minimizig the term $\sigma_k(\mathbf{s}_{k,})^2$ with respect to the $k$-th sensor angle. . . . . 44

2.6 Comparison of proposed solution to numerical/evolutionary techniques. . . . . 46

2.7 Comparison of two suboptimal solutions to GA over varying target-interference separation. . . . . 48

2.8 Comparison to GA over larger target-interference separations. . . . . 49

2.9 Optimal sensor configurations for two targets computed with Genetic Algorithm. . . 57

2.10 Sample trajectories of simulation scenario. . . . . 59

2.11 Error plot of the first simulation scenario. . . . . 60

2.12 Target trajectory plot for the second scenario. . . . . 61

2.13 Cross-Target statistics for Target 1 in scenario 2. . . . . 62

2.14 MSE of target 1 for various $\sigma_v$ in scenario 2. . . . . 62

2.15 Two target and estimate trajectory plot for full MASCOT with $\Phi = 30$. Filled positions marks where joint estimation takes place. . . . . 64

2.16 Log ratio of objective function for the two possible partitions when $\Phi = 10$ and $\Phi = 30$. 64

2.17 RMSE Performance Comparison between MASCOT adaptive filter and basic joint/separated estimation of both targets.. . . . . 65

2.18 Six Target Trajectory for Full Mascot Simulation. . . . . 66

2.19 Inter-target distance as a function of time in the six-target scenario. Different patterned/colored lines each represent a different target. . . . . 68

2.20 RMSE of Six-Target Full MASCOT Scenario with Comparison to Non-Adaptive Cooperative Filter. . . . . 69

3.1 Single Realization Trajectory Plot for the Interference Compensated Tracker . . . . 76
3.2 Norm Error Plot for a Single Realization of the Interference Compensated Tracker . 76
3.3 Measurement Bias Component being tracked for the first sensor . . . . . 77
3.4 RMSE over 100 trials for a synthetic bias with various $\sigma_e$ . . . . . 77
3.5 10 Real Interference Sources Deployed From the Target Location . . . . . 78
3.6 RMSE Over 100 Trials for the 10-Interferer Scenario . . . . . 78

4.1 (a) "True" asynchronous method (b) "False" asynchronous method (c) asynchronous sequential method (d) asynchronous batch method . . . . . 83
4.2 Time offset illustration for 3 asynchronous simulation scenarios . . . . . 87
4.3 Baseline RMSE performance for different sets of measurement times . . . . . 88
4.4 Performance for scenario 1 with asynchronous-compensation algorithms . . . . . 88
4.5 Normalized run-time as a function of the total number of particles . . . . . 89

5.1 Efficient Bias Function $b(\tau)$ for estimation of $g(\tau) = \frac{1}{\tau}$ with $\alpha = 10$, $\beta = 10$ . . . . . 104
5.2 Efficient Bias Function $b(\tau)$ for estimation of $g(\tau) = \frac{1}{\tau}$ with $\alpha = 10$, $\beta = 10$ rescaled in terms of $\frac{1}{\tau}$ . . . . . 105
5.3 Realization specific MSE for estimation of $\frac{1}{\tau} = \sigma^2$ with N = 1 sample, $\alpha = 10$, $\beta = 3$ 105
5.4 Realization specific MSE for estimation of $\frac{1}{\tau} = \sigma^2$ with N = 20 samples, $\alpha = 10$, $\beta = 3$ . . . . . 106
5.5 Ratio of $\frac{MSE}{CRLB}$ for various N, $\alpha = 10$, $\beta = 3$ . . . . . 106
5.6 Ratio of $\frac{MSE}{CRLB}$ for various N, $\alpha = 3$, $\beta = 3$ . . . . . 107
5.7 Minimum value of MSE/CRLB ratio . . . . . 108
5.8 Width of super-efficiency interval for $\beta = 3$ . . . . . 108
5.9 Efficiency of Bayes estimator as $\sigma \to \infty$ . . . . . 109
5.10 Single trajectory generated from the model with $\sigma_\epsilon = 0.1$ and $\sigma_\varsigma = 0.1$ . . . . . 121
5.11 MSE of trajectory in figure 5.10 with $\sigma_\epsilon = 0.1$ and $\sigma_\varsigma = 0.1$ . . . . . 122
5.12 |Experimental - Theoretical| MSE for trajectory in figure 5.10 with $\sigma_\epsilon = 0.1$ and $\sigma_\varsigma = 0.1$ . . . . . 122
5.13 First location component of bias function for trajectory in figure 5.10 with $\sigma_\epsilon = 0.1$ and $\sigma_\varsigma = 0.1$ . . . . . 123
5.14 Logarithmic plot of $\| \mathbf{b}_T^\star(\mathbf{X}_{1:T}) \|$ for various assumed values of $\sigma_\epsilon$ with all trajectories generated using fixed $\sigma_\epsilon = 10^{-6}$ . . . . . 123

5.15 PCRB plotted for various $\sigma_\epsilon$. Heavy lines are the PCRB, thin dotted lines are the MSE for specific trajectories generated from the model. . . . . 124
5.16 PCRB and MSE of the maneuvering trajectories plotted for various $\sigma_\epsilon$. Heavy lines are the PCRB, thin lines are trajectory-specific MSE. The assumed $\sigma_\epsilon$ matches actual $\sigma_\epsilon$ of the trajectories. . . . . 125
5.17 Sample trajectories with their respective MSE for various $\sigma_\epsilon$. . . . . 125
5.18 PCRB and MSE of the maneuvering trajectories plotted for various assumed $\sigma_\epsilon$. Heavy lines are the PCRB, thin lines are trajectory-specific MSE. All trajectories were generated with fixed $\sigma_\epsilon = 10^{-4}$ . . . . . 126
5.19 Sample trajectories with their respective MSE for various assumed $\sigma_\epsilon$ which is fixed at $10^{-4}$ during trajectory generation. . . . . 126
5.20 Logarithmic plot of $\parallel \mathbf{b}^\star_T(\mathbf{X}_{1:T}) \parallel$ for trajectories generated with $\sigma_\epsilon = 10^{-4}$. This value is again varied within the assumed model dynamics. . . . . 127
5.21 Sample trajectories with $\sigma_\epsilon$ fixed at $10^{-4}$ and varied only within model dynamics, plotted alongside location components of the bias vector (arrows). . . . . 127

6.1 Sensor and Target-Estimate Scatter Plot for $[\sigma_1, \sigma_2] = [1, 0.1]$ and $\hat{\mathbf{l}} = [-0.2, 0.1]^\top$ . 136
6.2 Measurement Distribution Plot with $[\sigma_1, \sigma_2] = [1, 0.1]$ and $\hat{\mathbf{l}} = [-0.2, 0.1]^\top$ . . . . . 136
6.3 Sensor and Target-Estimate Scatter Plot for $[\sigma_1, \sigma_2] = [1, 0.1]$ and $\hat{\mathbf{l}} = [1, 2]^\top$ . . . . 137
6.4 Measurement Distribution Plot using Solution 1 with $\epsilon = 0.1$,$[\sigma_1, \sigma_2] = [1, 0.1]$ and $\hat{\mathbf{l}} = [1, 2]^\top$ . . . . . 137
6.5 Measurement Distribution Plot using Method 1 with $\epsilon = 0.1$,$[\sigma_1, \sigma_2] = [1, 0.1]$ and $\hat{\mathbf{l}} = [1, 2]^\top$ . . . . . 138
6.6 Sensor and Target-Estimate Scatter Plot for $\epsilon = 0.1$,$[\sigma_1, \sigma_2] = [1, 0.01]$ and $\hat{\mathbf{l}} = [-1, -2]^\top$ . . . . . 139
6.7 Measurement Distribution Plot using solution 2 with $\epsilon = 0.1$,$[\sigma_1, \sigma_2] = [1, 0.01]$ and $\hat{\mathbf{l}} = [-1, -2]^\top$ for various values of $N$ . . . . . 140
6.8 Gaussian approximation of $f\left(y_{\mathbf{l}} \mid \hat{\mathbf{l}}, \hat{\mathbf{C}}\right)$, where the solid line is the true density and the dashed line is the approximation of the density. . . . . 141
6.9 Sensor and 6 Target-Estimate Scatter Plot for $\epsilon = 0.1$, $\sigma = [0.2, 0.1, 0.3, 0.2, 0.1, 0.3]$ and $r = 3$ . . . . . 143
6.10 Sensor and 6 Target-Estimate Distribution Plot for $\epsilon = 0.1$, $\sigma = [0.2, 0.1, 0.3, 0.2, 0.1, 0.3]$ and $r = 3$ . . . . . 143
6.11 Sensor and 6 Target-Estimate Scatter Plot for $\epsilon = 0.1$, $\sigma = [1.0, 0.5, 1.5, 1.0, 0.5, 1.5]$ and $r = 3$ . . . . . 144

6.12 Sensor and 6 Target-Estimate Distribution Plot for $\epsilon = 0.1$, $\sigma = [1.0, 0.5, 1.5, 1.0, 0.5, 1.5]$ and $r = 3$ . . . 144
6.13 Sensor and 6 Target-Estimate Scatter Plot for $\epsilon = 0.1$, $\sigma = [0.2, 0.1, 0.3, 0.2, 0.1, 0.3]$ and $r = 1$ . . . 145
6.14 Sensor and 6 Target-Estimate Distribution Plot for $\epsilon = 0.1$, $\sigma = [0.2, 0.1, 0.3, 0.2, 0.1, 0.3]$ and $r = 1$ . . . 145
6.15 Two Sensor, Single Target Measurement Geometry . . . 150
6.16 2 Sensor 1 Target Scatter Plot for $\mathbf{q}_1 = [0, 0]^\top$, $\mathbf{q}_2 = [-1, -2]^\top$, $\bar{\mathbf{l}} = [1, 1]^\top$, $\epsilon = 0.5$, $\Phi = 10$, and symmetric estimate covariance $\sigma = 0.2$ . . . 153
6.17 Measurement Scatter Plot with marginal histograms for 2-Sensor Single Target Scenario . . . 153
6.18 Distribution Plot for 2-Sensor Single Target Scenario . . . 154
6.19 Second View of Distribution Plot for 2-Sensor Single Target Scenario . . . 154
6.20 Two Sensor, Single Target Measurement Distribution Showing the Effects of Sensor Noise . . . 155
6.21 Two Sensor, Two Target Scatter Plot with $\mathbf{s}_1 = [0, 0]$, $\mathbf{s}_2 = [-1, -2]$,$\epsilon = 0.1$, $\Phi = 10$, $\zeta_k = 0.01$ . . . 157
6.22 Two Sensor, Two Target Distribution Plot with $\mathbf{s}_1 = [0, 0]$, $\mathbf{s}_2 = [-1, -2]$,$\epsilon = 0.1$, $\Phi = 10$, $\zeta_k = 0.01$ . . . 158
6.23 Two Sensor, Six Target Scatter Plot . . . 158
6.24 Two Sensor, Six Target Distribution Plot . . . 159

# List of Tables

2.1 Basic PF Algorithm for tracking a single target . . . . . 17
2.2 MASCOT Multi-Target Tracking PF Algorithm Implementation Summary . . . . . 36
2.3 Fast-Piecewise Algorithm Summary . . . . . 45

3.1 PF Algorithm for tracking a single target with static interference compensation . . . 72
3.2 PF Algorithm for tracking a single target with unknown interference modeled as a dynamically varying bias. . . . . 74

5.1 Recursive Algorithm for Realization-Specific MSE Computation at time $T$. . . . . 119

# Acknowledgements

I would like to give many thanks to my advisers Petar Djurić and Mónica Bugallo. I began my academic career at Stony Brook attending their engaging courses, which were my initial source of inspiration in following the path I chose. I will always remember them wholeheartedly entertaining all the questions I posed during class with answers that provoked even more thought and curiosity within me. I want to thank them for the countless hours they spent meeting with me throughout the years, providing just the right amount of guidance in keeping me on a sensible research path while leaving me with the creative freedom I believe is truly necessary for one to have in order to mature as a researcher. I must also thank them for the seemingly infinite patience they exercised in considering even my wildest of research ideas, always providing constructive feedback but also allowing me to be the judge of what was truly worthwhile to pursue and what was better left alone. I am truly grateful for having had them as advisers, not only were they instrumental in my end result, but more importantly, in nurturing the process of reaching maturity as an independent researcher.

Many thanks also to my committee members John Murray and Eugene Feinberg for taking the time to participate in the process. Professor Murray has helped me throughout the years in answering various theoretical questions and engaging me in interesting discussions (many in fact related more to my work as an Electrical Engineer than to my graduate studies). Professor Feinberg's challenging Dynamic Programming class allowed me to recognize the potential in viewing problems on a more theoretical (rather than intuitive) basis. I would also like to thank my classmates Cagla Tasdemir, Shishir Dash, and Iñigo Urteaga. I will always remember the endless hours Cagla and I spent in '256 studying for the Qualifying Exams, the meaningful discussions and exchange of ideas Shishir and I had, and the ICASSP conference adventures in Italy that Iñigo and I shared.

# 1

# Introduction

This dissertation provides a presentation of Multi-Agent Systems for Cooperative Tracking and details theoretical developments and experimental results obtained. As the specific nature of the thesis addresses multiple related topics, the dissertation is divided into separate sections addressing each item individually.

The main focus of the thesis deals with sequential estimation/tracking of a dynamic, high-dimensional state. While a major goal of this work is to provide a viable framework that can be directed towards high-dimensional systems in any applicable setting, much of the research has been done in a problem-specific context that is of particular interest to the author. It is in fact, the particular application considered which provides the underlying motivation for any results obtained. As such, the majority of this dissertation will make specific reference to the problem at hand.

## 1.1 Main Problem Description

The problem considered involves a system consisting of a large number of moving point-targets. It is desired to estimate the state of all targets (position, velocity, etc.) at each point in time based on some set of observations taken at that time and at times preceding it; this can be identified as a filtering problem in statistical estimation. A linear model is assigned to the motion of each target and it is further assumed that the targets do not interact with one another. The system described thus far does not necessarily pose a problem with dimensionality; in many cases the assumption of non-interaction allows targets to be treated as isolated entities, enabling an effective partitioning of the environment into multiple single-target systems. It is the nature of the observations which complicates the problem, as any observation taken is assumed to have some dependence on the state of all targets within the system. This can present a significant challenge with dimensionality. Since each observation is coupled to all states within the system it is no longer possible to directly divide the problem into isolated single-target tracking scenarios.

A system possessing the properties discussed, with an additional structure imposed on the observations, is of particular interest here. Specifically, it is assumed that there is some control over how each measurement is taken. This can be stated in another way: the *form* of the transformation from the target space to the observation space can be manipulated to some extent. Armed with this additional structure on the problem, it becomes possible to combat the obstacle of high dimensionality by seeking target-to-observation transformations admitting *sparse* representations. This is a key concept underlying the work presented and will be discussed in further detail within later sections.

A specific problem environment is now introduced which will be seen to possess all the qualities of the more general system just discussed. The environment consists of a number of targets to be tracked that are moving within a two-dimensional plane, each transmitting a constant signal of fixed strength. Observations, or more appropriately coined measurements in this context, are taken using sensors also located in the target space, which receive a portion of the transmitted signal energy from each target. The received signal strength at a sensor from an isolated target is inversely related to the spatial distance between the sensor-target pair. The net signal received from multiple targets at a given sensor is an *additive* combination of individual signal components. This measurement model can be seen to fall under the category of RSS sensing; a detailed mathematical formulation of the model will be presented later, the important concept to note here is that the level of coupling between each sensor and a given target is dependent on the spatial configuration of the two when a measurement is taken. All sensors within the environment are mobile and may have some limited awareness of their own locations. Additionally, each sensor is equipped with the ability to communicate with other sensors that currently reside within a neighboring region. Implementation details regarding the means for mobility, location-awareness, and inter-sensor communication are not specifically considered here; it is simply assumed the capability has been previously established based on currently existing technology.

### 1.1.1 MASCOT

It should now be obvious that the application-specific environment presented, which can be categorized as a mobile-sensor-network, matches well with the more general system initially described. While each measurement is affected by every target, sensors have the ability to position themselves in such a way so as to control the instantaneous correlation between a measurement and each target in the scene. It will be shown how this concept, coupled with the communication abilities of each sensor, allows for a feasible *distributed* solution to this inherently high-dimensional problem, using a divide-and-conquer approach. But first the concept of an *agent* in this specific context is introduced. A single agent represents a grouping of some number of sensors within the environment, acting as a unit, to accomplish a common estimation task. Associated with each agent is an

entity of computational resource that enables this task to be completed. The *sensor grouping is dynamic* and can change over time as the estimation environment evolves. In this fashion, one can then envision a network of these abstract agents, at any given time each tasked with estimating a portion of the target space, via the measurements from the sensors within their group, along with communicated information concerning the remaining space of the system, generated by the other agents in the network. Agents can manipulate the locations of their own sensors and determine what received information is relevant to completing their own task, in order to optimize performance. Development of this framework, specifically geared towards a system employing RSS sensors, is the main idea of the thesis, and is termed (M)ulti-(A)gent (S)ystems for (CO)operative (T)racking.

## 1.2 Detail-Oriented Challenges

Prior to work on the main goal of the thesis, a number of auxiliary topics were investigated relating to the practical implementation details of a mobile sensor network responsible for tracking a target. These topics fall under two main concepts and are discussed in what follows.

### 1.2.1 Measurement Interference

The situation with one or more sources of interference corrupting the measurements received by sensors was considered within a setting similar to the initial problem description, but with the problem reduced to a single-target tracking scenario. Initially the interference was assumed to originate from a single static source and an algorithm has been developed to compensate for this interference by localization of the source via a Rao-Blackwellized Particle Filter. A conference paper [1] was published for ICASSP 2011 presenting these results within a multi-sensor non-centralized target tracking environment. This was later extended to handle multiple, mobile interference sources with unknown dynamics. A general model was constructed to represent the net effect these "interferers" exert on the sensor measurements and an approach was formulated again within a Rao-Blackwellized PF framework to allow successful tracking to be maintained. The main results for this work were published [2] in the EUSIPCO 2013 conference. The latter extension can in some sense be seen as a very similar problem as the one initially outlined in section 1.1, if the interference sources are viewed as separate "foreign" targets. This can actually be a more challenging, and less approachable problem than when employing MASCOT, since in this case there is only a single agent; no external information has been received regarding the interferers and the agent must make inference regarding their influence on the measurements based on its sensors alone. It is believed that the approach outlined in [2] (and discussed in further detail in a later section) can be a valuable

supplement to MASCOT, particularly in the early phase of tracking where valid estimates have not yet been established for most agents.

### 1.2.2 Asynchronous Sensor Measurements

Within any sensing network collecting measurements that are *time-sensitive*, care must be taken when extracting information about a target to be estimated based on measurements originating from different sources. There is no guarantee these measurements were taken at the same time instants, and thus they will necessarily reflect different states of the target. Blindly assuming the target is in the same state for each measurement (corresponding to perfect synchronization of the measurement time instances) can result in significant performance degradation of any tracking algorithm. There exist several options for dealing with this issue; the most obvious being the explicit attachment of a timestamp to each transmitted measurement. Knowing the instants at which each measurement was taken allows the unit responsible for "measurement fusion" to easily make appropriate adjustments that will compensate for this asynchronism. However, there is additional communication overhead associated with timestamping the measurements and there is also the possibility of error within the timestamps themselves. Another option that is here termed "receiver timestamping" involves an explicit assumption made by the fusion center; namely that measurements are transmitted some fixed time after they are taken. With awareness regarding distances between the fusion center and transmitting sensors, along with the current time, the fusion center can form a reasonable estimate of when the measurement was made immediately upon its reception. However, neither of these options fully resolve the greater problem at hand to now be described. The fusion center represents the estimated target trajectory as a finite sequence of states at discrete instants in time. Assuming perfect synchronization between each set of measurements received from multiple sensors, the most natural approach would be to compute an estimate of the target state at exactly these measurement instants. New estimates are computed for these instances immediately upon reception of the full measurement set. However, this is not possible in the case of asynchronous measurements; there is now ambiguity as to which time instances should be used in representation of the estimated trajectory. One option, which is conceptually the simplest, is to update the estimated trajectory each time a new measurement is received. Essentially, this treats a multi-sensor tracking problem as a single-sensor problem with non-uniform (and possibly non-deterministic) time intervals between each estimate of the target. This can be a viable solution if the number of sensors in the system is small and if there is a reasonable spacing between each measurement instant. However, it becomes impractical if either of these conditions are not satisfied; small spacing between measurements requires a prohibitively high processing speed if the tracking is to be maintained in real-time, since the fusion center must have completed a target estimate update for the previous measurement before it receives a new one, or be forced to queue it until it is done processing. If the number

of sensors in the system is too large, processing may not be able to keep up with the incoming measurement stream regardless of the queue size. Even if the spacing is large relative to processor speed, it may still be desirable to minimize computational cost of the tracking algorithm. A more efficient alternative is to utilize a "batch" style method of processing to update the target estimate at fixed time intervals using any measurements that have been received since the last update. Two styles of algorithms implementing this method along with a comparison of performance and efficiency results against the "sequential" method just mentioned have been developed and were presented in a conference paper [3] for ICASSP 2012. The algorithms were again developed in the context of mobile distributed sensor networks employing PF algorithms to track a single target.

## 1.3 Performance Metrics and Underlying Bounds

As part of initial investigations, an attempt was made to compute performance bounds (particularly on the mean-square-error, or MSE) for several of the detail-oriented algorithms discussed in section 1.2. It quickly became apparent that for the specific RSS measurement model considered, conventional Bayesian bounds do not necessarily provide a sufficient characterization of the best attainable performance. To understand why, consider a simple example consisting of a single RSS sensor attempting to track a single target. Assuming the sensor measurements are corrupted by independent and stationary noise, the instantaneous SNR of each measurement is dependent solely on the actual distance between the sensor and target. As such, actual performance of a tracking algorithm will vary greatly depending on the specific trajectory realized by the target (which is assumed to follow a non-deterministic dynamic model). Bayesian bounds do not allow for this dependence; they reveal to us only the best possible "expected" performance of an estimator for random target trajectories generated under the dynamic model (prior) and random observations generated according to the measurement model, or the Bayesian Mean Square Error (BMSE). There is no information available in this bound indicating the best performance attainable for a specific trajectory realization, as the BMSE represents an average over all possible trajectories. Yet this information can be of considerable value as it can indicate if there are certain target trajectories which can lead to considerably poor (or excellent) performance. Essentially, this information can tell us how "sensitive" the estimation problem will be due to variations in the trajectory realization. In some instances it is postulated that this information can be used to manipulate the estimation environment in such a way that will produce a more favorable outcome, i.e., an obvious example in the simple scenario that was mentioned would be to position the sensor as close as possible to the target in order to maximize the measurement SNR.

Characterization of this best-possible "realization-dependent" performance was investigated at

great length in a generalized random-parameter estimation setting. Some informative theoretical results were obtained and are presented here.

# 2

# Dynamic State Estimation with MASCOT

## 2.1 Overview

The overall concept of dynamic state estimation is deceptively simple. At the most basic level, there is some unknown quantity we wish to estimate, or *infer the value of*, based on measurements we make which are correlated with it. Our unknown is not generally constant, but instead varies over time in some possibly non-deterministic way. We may also have some information regarding the dynamic behavior of the unknown that we may use in conjunction with our measurements to form a better inference. Our most logical option for inference at this point is to strike a balance between the information regarding the unknown we obtain from our measurements that is manifested via correlation, how we believe the state will evolve over time based on its dynamics, and the values we believe it has assumed before and after the point in time at which we wish to make inference. This mindset is essentially the core of Bayesian dynamic estimation, and the aforementioned "balanced inference" is embodied within our posterior probability density function of the unknown.

To be a bit more precise, we have a random scalar state $x_t$ that varies over discrete units of time $t$. We also have a set of random scalar measurements, $y_t$, that were taken at each point in time from $t = 1$ to $t = K$. Then our optimal inference of $x_t$ with respect to some loss criteria, will be based on $f(x_t|y_{1:K})$, the posterior *probability density function* (pdf), of $x_t$ conditioned on our set of measurements. Without delving into specific assumptions on $x_t$ at the moment, it is noted that any information regarding the state dynamics must be embedded within the posterior. This is made clear (under Markovian assumptions on the state) through factorization of the posterior, which admits some terms dependent only on "neighboring" values of $x_t$ in time $(x_{t-1}, x_{t+1})$, and not on the sequence of measurements. It should be noted at this point that the scope of this research has been limited to the filtering problem, or estimation of $x_t$ based on $y_{1:t}$, and no further reference will

be made to smoothing ($x_t$ from $y_{1:K}$ with $K \geq t$) or prediction ($K \leq t$). Although the Bayesian estimation framework indeed offers the promise of optimal inference in the form of the posterior, this is rarely an easily attainable object. Even for the simple case of a scalar state, the posterior is frequently mathematically intractable, and one must resort to numerical methods of computation or fall back upon suboptimal means of inference. Nonetheless, there do exist a large number of effective computational methods for dealing with this intractability; Monte Carlo simulation methods, particularly Particle Filtering approaches, have been widely accepted as a viable solution for sequential estimation problems, see [4] and [5] for a comprehensive treatment of the subject. It will be seen that particle filtering does play a key role in the specific implementation of MASCOT considered here.

As the dimensionality of $x_t$ grows however, the severity of the tractability situation is rapidly magnified; the computational cost of numerical evaluation rises exponentially and the performance of suboptimal techniques, particularly particle filtering [6], can suddenly deteriorate. This is not a unique phenomenon for statistical estimation and in fact occurs in a wide range of mathematical problems; this issue was originally coined the "Curse of Dimensionality" by Richard Bellman in 1961 when working with dynamic programming optimization problems. Regardless of this issue, there are a large class of important application-specific problems which indeed posses unknown states carrying a large number of dimensions. In [7], a Bayesian Monte Carlo approach is used to estimate hundreds of parameters of an ODE modeling the Janus Kinase/Signal Transducer and Activator of Transcription (JAK/STAT) pathway, which represents a major mechanism for chemical signaling through a biological cell membrane. Another example can be found in [8] where a form of particle filtering is used to estimate the failure rate of a Static Random-Access-Memory (SRAM) yield which is influenced by a large number of variation sources. Multiple target tracking (which is a focus in this thesis), feature tracking, facial recognition, and signal identification are all representative of high-dimensional dynamic state estimation with a highly active research community. Continuous advances in technology constantly reveal new applications where this obstacle is faced, making feasible solutions increasingly desirable.

It should be recognized that in the most general case, the dynamic filtering problem for even a scalar state is inherently a high-dimensional problem; each additional observation can be seen as representing another dimension of the posterior and thus the dimensionality grows linearly over time. However, this does not hold true when a simple restriction is made on the class of system dynamics considered. Specifically, those systems whose state value at time $t$ depend only on a finite number of past states, do not generate indefinite growth in dimension for the filtering problem. In fact, a system possessing the Markovian property, whereby the current value of the state depends only on its previous value, admits a filtering solution that remains constant in dimensionality. To

clarify what is meant by this, consider the well-known decomposition of the posterior distribution made possible under the Markovian assumption and with assumed conditionally independent observations ($y_t$ depends on $x_t$ only):

$$f(x_t|y_{1:t}) \propto f(y_t|x_t)f(x_t|y_{1:t-1}) = f(y_t|x_t)\int f(x_t|x_{t-1})f(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{2.1}$$

Here the posterior is proportional to the product of the observation likelihood, $f(y_t|x_t)$, with dimension 1, and the predictive density $f(x_t|y_{1:t-1})$ with dimension $t-1$[1]. Thus while the overall posterior remains as a $t$-dimensional function, its computation can be effectively partitioned into the product of two uncoupled functions with lesser dimension; performing further recursions on $f(x_t|y_{1:t-1})$ enable the effective dimensionality of the problem to be reduced to that of $x_t$ itself.

Notice that this reduction in dimensionality is a direct result of the assumed *sparsity* of the problem; that $x_t$ is dependent only on a localized region of its history. This is in fact an observation which can be applied in a very general sense, to any problem of high dimension. When the structure of a system admits a sparse representation, its exploitation can be used to effectively combat the problem of dimensionality by transforming the system to a space of reduced dimension. A clear example of this dimension reduction can be found in a long-standing methodology [9] for large-scale multivariate analysis, called Principal Components Analysis (PCA). Essentially, this method approximates a large matrix of data $\mathbf{X}$ as the product of two smaller matrices, $\mathbf{P}'$ and $\mathbf{T}$; this can be viewed as projecting the data down to a lower-dimensional subspace via the projection matrix $\mathbf{P}'$ with the matrix $\mathbf{T}$ describing the coordinates within this hyperplane. The amount by which the dimension can be reduced using this method depends directly on the number of "principal" components that are needed to sufficiently describe the data; the sparser the underlying system actually is, the smaller the resulting dimensionality will be.

## 2.2 The general idea of MASCOT

Leveraging sparsity in a high-dimensional state space is the backbone of the MASCOT framework, which will now be presented in a general sense then later specialized to the application of interest. In order to avoid notational ambiguity, let us first form a specific construction of the space $\mathbb{R}^L$. To do this, form $L$ *distinct* copies of the real-number set $\mathbb{R}$ as $\mathbb{R}_1, \mathbb{R}_2, \ldots, \mathbb{R}_L$. Recall the definition of a Cartesian product: for two sets $\mathbf{A}$ and $\mathbf{B}$, the product $\mathbf{A} \times \mathbf{B}$ is the set of all *ordered pairs* $(a, b)$ where $a \in \mathbf{A}$ and $b \in \mathbf{B}$. We can then construct $\mathbb{R}^L$ as,

$$\mathbb{R}^L = \mathbb{R}_1 \times \mathbb{R}_2 \times \cdots \times \mathbb{R}_L = \prod_{l=1}^{L} \mathbb{R}_l \tag{2.2}$$

[1]Notice that here the dimensionality of the function is meant to reference the number of input scalar arguments we are conditioning on. The state itself ($x_t$) is assumed to be scalar.

An element of $\mathbb{R}^L$ based on this construction as then an *ordered n-tuple* $\tilde{x} = (x_1, x_2, \ldots, x_L)$, subsequently written as a column vector (but where it is understood that a specific ordering has been proscribed to the space). Equipped with this construction we can form a distinct subspace of $\mathbb{R}^L$ by way of an ordered *index set* $\mathbb{S} \subset \{j : j \in 1, 2, \ldots, L\}$. Namely, we can define the subspace $\mathbb{X}$ as,

$$\mathbb{X} = \prod_{j \in \mathbb{S}} \mathbb{R}_j \tag{2.3}$$

The *projection function* $\mathrm{proj_i}(\tilde{\mathrm{x}})$ takes the element $\tilde{x} \in \mathbb{R}^L$ as input and returns the $i$th component, i.e., $\mathrm{proj_i}(\tilde{\mathrm{x}}) = \mathrm{x_i}$. With this in mind, we can then write $\mathrm{proj}_{\{\mathbb{S}\}_\mathrm{i}}(\tilde{\mathrm{x}}) = \mathrm{x}_{\{\mathbb{S}\}_\mathrm{i}}$ where $\{\mathbb{S}\}_i$ is the $i$th element in $\mathbb{S}$. It should then be clear from the preceding definition that for every element $\tilde{x}$ in $\mathbb{R}^L$ there is a corresponding element in $\mathbb{X}$ that can be written as a $|\mathbb{S}| \times 1$ vector $= \begin{bmatrix} x_{\{\mathbb{S}\}_1} x_{\{\mathbb{S}\}_2} & \ldots & x_{\{\mathbb{S}\}_{|\mathbb{X}|}} \end{bmatrix}^\top$. As an example, assume we have the vector $\mathbf{X} = \begin{bmatrix} 3.2 & 1.2 & -.12 & 2.1 & 8.7 \end{bmatrix}$ in $\mathbb{R}^5$ and the index set $\mathbb{S} = \{1, 4\}$. The corresponding element in $\mathbb{X}$ is then $\begin{bmatrix} 3.2 & 2.1 \end{bmatrix}$.

Let us then assume we wish to estimate an $L$-dimensional state at time $t$, $\mathbf{X}_t \in \mathbb{R}^L$, using $L$ agents, each with their own set of "private" measurements of the state (here it is assumed measurements are not shared across agents). The $i$-th agent directly estimates the vector $\mathbf{X}_{\mathbf{S}_{i,t}}$ in the subspace $\mathbb{X}_{\mathbf{S}_{i,t}}$, of the full state space, with the remainder of the state, $\mathbf{X}_{\mathbf{F}_{i,t}} \in \mathbb{X}_{\mathbf{F}_{i,t}}$, treated as nuisance parameters. Notice the symbols $\mathbf{S}_{i,t}$ and $\mathbf{F}_{i,t}$; these are indexing sets used to identify the specific dimensions of $\mathbb{R}^L$ from which the subspaces associated with agent $i$ are constructed. Namely, these sets are defined as:

$$\begin{aligned} \mathbf{S}_{i,t} &= \left\{ j : [\mathbf{X}_t]_j \text{ estimated by agent } i \right\} \\ \mathbf{F}_{i,t} &= \mathbf{S}_{i,t}^c \end{aligned} \tag{2.4}$$

where here $[\mathbf{X}_t]_j$ refers to the $j$-th component of $\mathbf{X}_t$. Using these sets, the subspaces associated with agent $i$ are constructed using the Cartesian product as follows:[1]

$$\mathbb{X}_{\mathbf{S}_{i,t}} = \prod_{j \in \mathbf{S}_{i,t}} \mathbb{R}_j \qquad \mathbb{X}_{\mathbf{F}_{i,t}} = \prod_{j \in \mathbf{F}_{i,t}} \mathbb{R}_j \tag{2.5}$$

Thus here it can be stated that $\mathbb{X}_{\mathbf{S}_{i,t}} = \mathbb{R}^{|\mathbf{S}_{i,t}|}$ where $|\mathbf{S}_{i,t}|$ is the cardinality of $\mathbf{S}_{i,t}$, although the *ordering* of $\mathbb{X}_{\mathbf{S}_{i,t}}$ is distinct, while for $\mathbb{R}^{|\mathbf{S}_{i,t}|}$ it is not. The full state-space is thus *partitioned* amongst all the agents, implying that:

$$\prod_{i=1}^{L} \mathbb{X}_{\mathbf{S}_{i,t}} = \mathbb{R}^L \tag{2.6}$$

[1] Keeping in mind the initial construction of $\mathbb{R}^L$ in (2.2)

The collection of all agent *in-focus* subspaces, $\mathbb{X}_{\mathbf{S}_{i,t}}$, which is labeled $\Theta_t$, can indeed be viewed as a partition of the state space at time $t$, i.e.,

$$\Theta_t = \left\{ \mathbb{X}_{\mathbf{S}_{1,t}} \;\; \mathbb{X}_{\mathbf{S}_{2,t}} \;\; \dots \;\; \mathbb{X}_{\mathbf{S}_{L,t}} \right\} \tag{2.7}$$

While agents do not directly estimate the vector $\mathbf{X}_{\mathbf{F}_{i,t}}$, they rely on information communicated from all other agents, labeled $\Psi\left(\mathbf{X}_{\mathbf{F}_{i,t}}\right)$ to compensate for any "interference" effect $\mathbf{X}_{\mathbf{F}_{i,t}}$ may have on the agent's own measurements. The external information is a generalized notion and can represent a number of different forms; the two most obvious being a point estimate $\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}$ or a probability density $f\left(\mathbf{X}_{\mathbf{F}_{i,t}}\right)$.

The partition $\Theta_t$ is *not fixed* but is instead dynamically chosen at each time $t$ as follows:

Choose $\Theta_t$ such that:

$$\mathcal{O}_t\left(\Theta_t\right) = \underbrace{\sum_{i=1}^{L} \mathbb{D}_M\left(\mathbb{X}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t}}\right)\right) \mathbb{B}_i\left(\mathbf{X}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t}}\right)\right)\Big|_{\substack{\mathbf{X}_{\mathbf{S}_{i,t}}=\hat{\mathbf{X}}_{\mathbf{S}_{i,t}} \\ \mathbf{X}_{\mathbf{F}_{i,t}}=\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}}} + \lambda_1 \mathbb{D}_A\left(\Theta_t\right)}_{\mathbb{B}(\Theta_t)} \tag{2.8}$$

is minimized

Where $\mathbb{B}_i$ is a measure of the "predicted accuracy" that agent $i$ will have in estimating $\mathbf{X}_{\mathbf{S}_{i,t}}$, evaluated at a pointwise prediction of $\mathbf{X}_{\mathbf{S}_{i,t}}$ ( labeled $\hat{\mathbf{X}}_{\mathbf{S}_{i,t}}$ and based on agent $i$'s own processing) along with a pointwise prediction of $\mathbf{X}_{\mathbf{F}_{i,t}}$ (labeled $\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}$) that is derived from $\psi\left(\mathbf{X}_{\mathbf{F}_{i,t}}\right)$. One natural choice for $\mathbb{B}_i$ would be to use an estimation error bound (such as the CRLB); this will be illustrated for the specific application considered in section 2.6. The functions $\mathbb{D}_M$ and $\mathbb{D}_A$ associate a cost with the dimensionality "spread" of the partition, i.e., one (rather arbitrary) example choice for $\mathbb{D}_A$ would be:

$$\mathbb{D}_A\left(\Theta_t\right) = \exp\left[\lambda_2 \max_{\mathbb{X}_{\mathbf{S}_{i,t}} \in \Theta_t} \dim\left(\mathbb{X}_{\mathbf{S}_{i,t}}\right)\right] \tag{2.9}$$

The scaling parameters $(\lambda_1, \lambda_2)$ can control the trade-off between estimation accuracy and dimensionality reduction. It is interesting to note however, that a higher dimensionality may not always correlate with better performance; this will be seen in an example scenario. The function $\mathbb{D}_M$ is of particular interest here; this associates a cost with the dimensionality of a *given* agents partition by *scaling* $\mathbb{B}_i$ and is considered significantly less arbitrary than $\mathbb{D}_A$. Just as the CRLB is a natural choice for $\mathbb{B}_i$, it is believed that one can develop an optimal choice for $\mathbb{D}_M$ based on the specific tracking algorithm. Some initial work has been done in this area regarding the specific application considered (see 2.6) leading to a function which has been found to perform well, however this remains an open area.

## 2.3 Introduction to the Multi-Target-Tracking Application

While the MASCOT algorithm was just introduced in a general setting, work has been focused on its implementation within a multi-target tracking (MTT) context using a network of mobile sensors. This specific tracking environment along with the nature of the dynamical and measurement models considered is what initially motivated development of the MASCOT framework.

The environment considered consists of $L$ moving targets that exist within a 2-dimensional plane. The state of the $l$-th target at time $t$ is denoted as $\mathbf{x}_{l,t}$ and can be decomposed as $\mathbf{x}_{l,t} = \begin{bmatrix} x_{l,t}^{[1]} & x_{l,t}^{[2]} & x_{l,t}^{[3]} & x_{l,t}^{[4]} \end{bmatrix}^\top$ where $(x_{l,t}^{[1]}, x_{l,t}^{[2]})$ represent the target's position coordinates and $(x_{l,t}^{[3]}, x_{l,t}^{[4]})$ represent its velocity coordinates. Motion of a target is modeled as:

$$\mathbf{x}_{l,t} = \mathbf{A}\mathbf{x}_{l,t-1} + \mathbf{w}_{l,t} \tag{2.10}$$

where $\mathbf{w}_{l,t}$ is a $4 \times 1$ vector representing the state process noise and is here assumed to be Gaussian distributed with zero mean and covariance matrix $\mathbf{Q}$ defined as:

$$\mathbf{Q} = \sigma_w^2 \begin{bmatrix} \frac{1}{3}T^3 & 0 & \frac{1}{2}T^2 & 0 \\ 0 & \frac{1}{3}T^3 & 0 & \frac{1}{2}T^2 \\ \frac{1}{2}T^2 & 0 & T & 0 \\ 0 & \frac{1}{2}T^2 & 0 & T \end{bmatrix} \tag{2.11}$$

$\mathbf{A}$ is termed the state transition matrix and is:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.12}$$

In eqs. (2.11) and (2.12) the parameters $\sigma_w^2$ and $T$ represent the state process noise intensity and the underlying time interval respectively. This model represents a discretization of a continuous-time, second order model that will be described here for a target moving in a single coordinate (it will be obvious how the derivation is extended to cover a 2D state). Let the variable $\tau$ here denote continuous time in contrast to the discrete variable $t$ which is used to represent a sampling instant of $\tau$, i.e., $\tau_t = tT$. Let $p(\tau)$ denote the position of the target, $v(\tau) = \frac{\partial p(\tau)}{\partial \tau}$ its velocity, and $a(\tau) = \frac{\partial v(\tau)}{\partial \tau}$ its acceleration which is assumed random. Namely, $a(\tau) = w(\tau)$ where $w(\tau)$ is white-noise with autocorrelation $R_w(\tau) = \sigma_w^2 \delta(\tau)$. The state equations for this model are then,

$$\mathbf{x}(\tau) = \begin{bmatrix} p(\tau) \\ v(\tau) \end{bmatrix}; \quad \frac{\partial}{\partial \tau}\mathbf{x}(\tau) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(\tau) \tag{2.13}$$

Equation (2.13) is of the form,

$$\frac{\partial}{\partial \tau}\mathbf{x}(\tau) = \mathbf{F}\mathbf{x}(\tau) + \mathbf{g}w(\tau) \tag{2.14}$$

It is straightforward to show that,

$$\mathbf{x}(\tau) = e^{\mathbf{F}(\tau-\tau_t)}\mathbf{x}(\tau_t) + \int_{\tau_t}^{\tau} e^{\mathbf{F}(\eta-\tau_t)}\mathbf{g}w(\eta)\,\partial d\eta \tag{2.15}$$

One can then develop the model in (2.10) directly from (2.15) with the relation,

$$\begin{aligned}\mathbf{A} &= e^{\mathbf{F}T} \\ \mathbf{w}_t &= \int_{\tau_t}^{\tau_{t+1}} e^{\mathbf{F}(\tau_{t+1}-\eta)}\mathbf{g}w(\eta)\,d\eta\end{aligned} \tag{2.16}$$

The covariance matrix of $\mathbf{w}_t$ is,

$$\begin{aligned}\mathbf{Q} = \mathbb{E}\left[\mathbf{w}_t\mathbf{w}_t^\top\right] &= \mathbb{E}\left[\left(\int_{\tau_t}^{\tau_{t+1}} e^{\mathbf{F}(\tau_{t+1}-\eta)}\mathbf{g}w(\eta)\,d\eta\right)\left(\int_{\tau_t}^{\tau_{t+1}} e^{\mathbf{F}(\tau_{t+1}-\eta')}\mathbf{g}w(\eta')\,d\eta'\right)^\top\right] \\ &= \int_{\tau_t}^{\tau_{t+1}}\int_{\tau_t}^{\tau_{t+1}} e^{\mathbf{F}(\tau_{t+1}-\eta)}\mathbf{g}\mathbf{g}^\top\left(e^{\mathbf{F}(\tau_{t+1}-\eta)}\right)^\top \mathbb{E}\left[w(\eta)\,w(\eta')\right]d\eta d\eta'\end{aligned} \tag{2.17}$$

Noting that $\mathbf{F} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}$ and $\mathbb{E}\left[w(\eta)\,w(\eta')\right] = \sigma_w^2\delta(\eta)$ in this specific model we then arrive at the forms embedded within (2.12) and (2.11),

$$\mathbf{A} = \exp\left(\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\right) = \begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix} \tag{2.18}$$

and,

$$\begin{aligned}\mathbf{Q} &= \int_{\tau_t}^{\tau_{t+1}}\int_{\tau_t}^{\tau_{t+1}} \begin{bmatrix}1 & \tau_{t+1}-\eta\\ 0 & 1\end{bmatrix}\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ \tau_{t+1}-\eta' & 1\end{bmatrix}\sigma_w^2\delta(\eta-\eta')\,d\eta d\eta' \\ &= \int_{\tau_t}^{\tau_{t+1}} \begin{bmatrix}(\tau_{t+1}-\eta)^2 & \tau_{t+1}-\eta\\ \tau_{t+1}-\eta & 1\end{bmatrix}\sigma_w^2 d\eta \\ &= \sigma_w^2\begin{bmatrix}-\frac{1}{3}(\tau_{t+1}-\eta)^3\big|_{\tau_t}^{\tau_{t+1}} & -\frac{1}{2}(\tau_{t+1}-\eta)^2\big|_{\tau_t}^{\tau_{t+1}}\\ -\frac{1}{2}(\tau_{t+1}-\eta)^2\big|_{\tau_t}^{\tau_{t+1}} & \eta\big|_{\tau_t}^{\tau_{t+1}}\end{bmatrix} \\ &= \sigma_w^2\begin{bmatrix}\frac{1}{3}T^3 & \frac{1}{2}T^2\\ \frac{1}{2}T^2 & T\end{bmatrix}\end{aligned} \tag{2.19}$$

All targets are assumed to continuously transmit a constant signal of fixed energy. Information is collected regarding the targets using sensors that receive an additive combination of all signals transmitted by the targets. A measurement taken by the $i$-th sensor at time $t$ is modeled as follows:

$$y_{i,t} = \sum_{l=1}^{L}\frac{\Phi}{\|\mathbf{q}_{i,t} - \mathbf{x}_{l,t}^{1:2}\|^\alpha + \epsilon} + v_{i,t} \tag{2.20}$$

where $\mathbf{q}_{i,t}$ and $\mathbf{x}_{l,t}^{1:2}$ are the position coordinates of the $i$-th sensor and the $l$-th target respectively, $\epsilon$ is a fixed saturation parameter which limits the total amount of signal power the sensor can receive from a given target, $\Phi$ is the transmitted signal power, $\alpha$ is a path-loss coefficient dependent on the physical transmission medium, and $v_{i,t} \sim \mathcal{N}(0, \sigma_v^2)$ is the sensor noise. All sensors are mobile

and capable of adjusting their positions to enhance signal reception.

Note the model described represents a specific flavor of the tracking problem that is applicable to a wide variety of applications incorporating RSS measurements, including RF and acoustic signal sensors. Work has been focused on tracking; it is assumed that valid measurements are received from all targets at each time instant (there are no "misses" or "false alarms") and that the total number of targets in the environment is known and constant over time (targets do not suddenly exit/enter the scene). While this does exclude a fair amount of applications (particularly those involving radar-scanning) it maintains the emphasis on track estimation rather than target-detection.

## 2.4 Basic Target Tracking Background

One of the earliest approaches to target tracking using Bayesian estimation principles involves incorporation of a Kalman-filter [10] to track the state of a single target based on one or more observations. The Kalman filter involves recursive computation of an estimated mean and covariance for the target's current state, based on the newly received measurement, along with the estimated mean and covariance of the target determined at the prior time step. It is well-known that the Kalman filter is optimal under the assumption of linear state dynamics and measurement models with Gaussian-distributed noise. This optimality does not hold for more general models and can in fact yield rather poor results when applying the Kalman-filter to a non-linear non-Gaussian problem. To combat nonlinearities within the models, the Extended Kalman Filter was developed which essentially carries out the Kalman Filtering steps on a linearization of the dynamic/measurement models about the current mean and covariance estimates. This extension can be effective for situations where the nonlinearities present are mild, but can deteriorate rapidly for more general models particularly those involving non-Gaussian noise. A more recent development called the Unscented Kalman Filter (UKF) [11] uses a set of deterministic points (sigma points) that are generated by representing the target state as a discrete distribution based on its estimated mean and covariance in the previous time. These points are propagated through the nonlinear dynamic/measurement models and the resulting transformed points are used within the conventional KF steps to determine an estimate for the target mean and covariance of the current time. While possibly more computationally intensive, the UKF has been found to yield superior performance to that of the EKF in most cases. However, difficulties still arise when dealing with systems containing non-Gaussian noise. A form of sequential Monte Carlo techniques, called particle-filtering, is able to more effectively deal with generalized models for the target motion and sensor measurements. Particle filtering uses a discrete random measure (or collection of randomly drawn particles and associated weights) to represent the posterior distribution of the target state at each point in time. While the ideal distribution to use for drawing the particles is the posterior itself, this is generally not

possible. Instead, the random measure is generated using a technique called importance sampling: particles are drawn from a "proposal" distribution that is easier to sample from than the posterior, and weighted in such a way that expectation over a random measure from this proposal corresponds to expectation over a random measure from the target posterior.

Since particle filtering plays a crucial role in the application-specific implementation of MASCOT discussed, its use will now be illustrated in detail for a simple case. Assume there is a single target, $\mathbf{x}_t$ which follows the model as outlined in section 2.3. The posterior distribution of the target state at time $t$, conditioned on all observations received up to the present time, is denoted as $f(\mathbf{x}_t|y_{1:t})$, and is approximated using a discrete representation consisting of $M$ particles as follows:

$$f(\mathbf{x}_t|y_{1:t}) \approx \sum_{m=1}^{M} w_t^{(m)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(m)}\right) \tag{2.21}$$

where $\delta(\cdot)$ is the Dirac-delta function, $x_t^{(m)}$ is the $m$-th particle, and $w_t^{(m)}$ is the weight of the $m$-th particle. Particles are generated from the proposal distribution, i.e. $\mathbf{x}_t^{(m)} \sim q(\mathbf{x}_t|y_{1:t})$, and it is desired to formulate an expression for the weights, $w_t^{(m)}$. Let us denote the expectation of a test function $h(\mathbf{x}_t)$ under the true posterior as $\mathcal{I}\left[h(\mathbf{x}_t)\right]$, or:

$$\mathcal{I}\left[h(\mathbf{x}_t)\right] = \mathbb{E}_f\left[h(\mathbf{x}_t)\right] = \int h(\mathbf{x}_t) f(\mathbf{x}_t|y_{1:t}) d\mathbf{x}_t \tag{2.22}$$

We also write $\hat{\mathcal{I}}\left[h(\mathbf{x}_t)\right]$ for the random variable generated by replacing $f(\mathbf{x}_t|y_{1:t})$ in (2.22) with its particle representation formed using the proposal,

$$\begin{aligned}\hat{\mathcal{I}}\left[h(\mathbf{x}_t)\right] &= \int h(\mathbf{x}_t) \left(\sum_{m=1}^{M} w_t^{(m)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(m)}\right)\right) d\mathbf{x}_t \\ &= \sum_{m=1}^{M} w_t^{(m)} h(\mathbf{x}_t^{(m)})\end{aligned} \tag{2.23}$$

It is then desired to choose the weights such that the expectation of $\hat{\mathcal{I}}\left[h(\mathbf{x}_t)\right]$ (under the proposal) forms an unbiased estimator of $\mathcal{I}\left[h(\mathbf{x}_t)\right]$.

$$\begin{aligned}\mathbb{E}_q\left[\hat{\mathcal{I}}\left[h(\mathbf{x}_t)\right]\right] &= \sum_{m=1}^{M} \mathbb{E}_q\left[w_t^{(m)} h(\mathbf{x}_t^{(m)})\right] \\ &= \sum_{m=1}^{M} \int w_t^{(m)} h(\mathbf{x}_t^{(m)}) q(\mathbf{x}_t^{(m)}|y_{1:t}) d\mathbf{x}_t^{(m)}\end{aligned} \tag{2.24}$$

From (2.24) it should be obvious that choosing

$$w_t^{(m)} = \frac{1}{M} \frac{f(\mathbf{x}_t^{(m)}|y_{1:t})}{q(\mathbf{x}_t^{(m)}|y_{1:t})} \tag{2.25}$$

yields the desired result. Since the ratio in (2.25) may be known only up to a multiplicative constant, the "true" weights in (2.21) are replaced with normalized weights,

$$\hat{w}_t^{(m)} = \frac{w_t^{(m)}}{\sum_{i=1}^{M} w_t^{(m)}} \tag{2.26}$$

yielding the relation $\hat{w}_t^{(m)} = c w_t^{(m)}$ with $c$ some unknown constant.

The most frequently chosen proposal distribution in a target-tracking context is the transition prior, $f(\mathbf{x}_t|\mathbf{x}_{t-1})$, since it is described by the target state dynamical model and in most cases is easy to sample from. It can be shown that this choice for the proposal yields the following recursive expression for the particle weights:

$$\hat{w}_t^{(m)} = \frac{\hat{w}_{t-1}^{(m)} f(y_t|\mathbf{x}_t^{(m)})}{\sum_{i=1}^{M} \hat{w}_{t-1}^{(i)} f(y_t|\mathbf{x}_t^{(i)})} \tag{2.27}$$

The main problem with the use of the transition prior as the proposal lies in the fact that $f(\mathbf{x}_t|\mathbf{x}_{t-1})$ has no dependency on the current measurement; there is no guarantee that sampled particles will fall in a region of high probability mass within the observation likelihood, $f(y_t|\mathbf{x}_t^{(m)})$. As a result, particle degeneracy can occur, whereby a large number of the particles have negligible weights; eventually the filter can "collapse" yielding only a single particle with non-zero weight. To combat this problem, a technique called "resampling" is employed whereby a new particle set is generated from the current weighted set. There are several ways to conduct resampling, each with their own benefits and drawbacks. The most basic form uses sampling-with-replacement; for each new particle, an index $k_i \in \mathbb{Z}$ within the range $[1, M]$ is drawn from the discrete distribution $\text{Prob}(k_i) = w_t^{(k_i)}$. The $i$-th particle in the new set is then set to $\mathbf{x}_t^{(i)} = \mathbf{x}_t^{(k_i)}$; it should be clear that the new set will consist of particles from the original set that tend to have higher weights. By resampling at appropriate times, there is a better chance that a larger number of particles will maintain non-zero weights, since they are "guided" into regions of higher mass within the observation likelihood. Resampling does not come without its own problems; due to the continued replication of high-probability particles over time, "sample-impoverishment" can occur whereby there is a very low-variance in the values of the particles themselves, i.e., the support of the posterior is not adequately covered by the particle set. Numerous solutions have been proposed to deal with this "degeneracy-impoverishment" tradeoff, the most well-known being to resample only when some measure of the particle diversity has fallen below a certain threshold. See [4] and [12] for a more comprehensive treatment of this subject.

Returning to the simple single-target example described, a full particle filtering algorithm based

| |
|---|
| *Step 1:* For $(m = 1 : M)$, draw $\mathbf{x}_t^{(m)} \sim \mathcal{N}(\mathbf{A}\mathbf{x}_{t-1}^{(m)}, \mathbf{Q})$ |
| *Step 2:* Compute $w_t^{(m)} = w_{t-1}^{(m)} \mathcal{N}\left[y_t \mid \frac{\Phi}{\|\mathbf{q}_t - \mathbf{x}_t^{(m)}\|^{\alpha} + \epsilon}, \sigma_v\right]$ |
| *Step 3:* Normalize weights by setting $\hat{w}_t^{(m)} = w_t^{(m)} / \sum_{i=1}^{M} w_t^{(i)}$ |
| *Step 4:* Generate target state estimate as $\tilde{\mathbf{x}}_t = \sum_{m=1}^{M} \hat{w}_t^{(m)} \mathbf{x}_t^{(m)}$ |
| *Step 5:* If necessary, resample particles by: |
| For $(i = 1 : M)$ |
| Set $\mathbf{x}_t^{(i)} = \mathbf{x}_t^{(k_i)}$ and $w_t^{(i)} = \frac{1}{M}$ where Prob $(k_i = j) = \hat{w}_t^{(j)}$ |

**Table 2.1:** Basic PF Algorithm for tracking a single target

on using the transition prior as the proposal, can be outlined for estimation of the target state at each point in time. Here it is assumed that tracking has been properly initialized with some well defined means to ensure the initial particle set represents a reasonable approximation to the initial target state posterior. Then update of the estimated target state at time $t$ proceeds as in Table 2.1.

Using a particle filter for this model may not be entirely necessary since the state and measurement noise are Gaussian, however the algorithm does handle well the nonlinearity present within the measurement model. Notice also that in this simple example, there is only one measurement captured from a single sensor each time instant. While it can be said in general that the use of a larger number of concurrent measurements will result in improved performance, it is particularly true in the case of RSS measurements. To see why, consider a single noiseless measurement of the target. For a given value of $y_k$, there is a circular locus with radius $r = \sqrt{\frac{\Phi}{y_t} - \epsilon}$, of possible target locations centered around the sensor location which could have produced this value. This represents a rather severe ambiguity and manifests itself as multi-modality within the observation likelihood. With two or three concurrent sensor measurements (assuming the measurements are not taken from co-located sensors) this ambiguity is immediately, and dramatically, reduced.

## 2.5 Cooperative-Agent Systems for Multi-Target Tracking

The concept of employing multiple autonomous agents to cooperatively handle a task of high complexity has become a widely studied topic [13]. Numerous applications abound across a diverse array of fields, including machine learning [14], network security management [15], medical data processing [16], and particularly sensor networks [17]. Utilization of a multi-agent system (MAS) within a target-tracking environment offers a number of benefits including improved reliability through redundancy, greater information processing capability, and network-induced enhancements in flexibility, control, and scalability.

The specific nature of a multi-target tracking problem lends itself well to a MAS with an underlying distributed processing format. As the number of targets within the environment grows, the computational requirements can quickly surpass what can be supported by a conventional centralized processing scheme. This is particularly true in the case of sensor networks which are mobile; a large amount of energy is necessarily expended on conveying individual sensor information back to the data fusion center, which can easily lie outside the physical range that would allow for direct communication if the span of the tracking environment is large enough. More importantly, there exists an important benefit that is realized by distributed processing in this context which is rather unique to the multi-target tracking problem. Namely, the structure of the underlying state-space model is such that it can be naturally partitioned into many smaller weakly correlated subspace representations. To further clarify this, consider the fairly degenerate situation where targets are spaced at a nearly-infinite distance away from one another. For the RSS-type measurements considered, a sensor positioned relatively close to one target will not be affected by the other targets; as a result the tracking of each target can be considered separately as individual single-target tracking problems. It is obvious in this case there exists the natural partition of the full state space, into near-independent subspaces each corresponding to an individual target.

This key concept, of exploiting a "distributed representation" of the target space in order to approach the tracking problem has been recognized previously and studied at great lengths in [18]. There, a solution is presented which adaptively switches between estimation of a number of targets within the full-dimensional "joint space", to dimensionally separated partitions each representing the marginal distributions or "marginal space" of individual targets. Determination of when to switch models is mainly based on the current estimated inter-target distances. A problem arises when switching from the joint to marginal space, which is done by marginalization of the joint distribution into individual target distributions; as no information is present within the sensor measurements that distinguishes one target from another, there will be multiple modes in all of the target marginals corresponding to each of the possible target configurations which could have produced a given value for the measurement. There is a solid focus on addressing this identity management problem within the paper that is elaborated further within [19]. Work in the previous two papers mentioned is quite similar to MASCOT, however there are many key differences between the aforementioned results and the MASCOT framework. A heuristic based on the instantaneous cross-target coupling within the sensor measurements is used to determine the switch between the joint/marginal state space representations in [18], whereas the adaptive state-space partitioning that takes place as part of MASCOT relies on optimization of an objective function as outlined in equation (2.8). This objective naturally seeks the sparsest partition (allowing for the greatest reduction in dimensionality) that can be realized without excessively compromising estimation accuracy. Another major difference which distinguishes MASCOT lies in the fact that

the partitions formed *never truly represent marginal distributions*; the estimating agent will always incorporate externally communicated probabilistic information regarding other target states into its own estimation process. It is also believed that MASCOT is better suited for a mobile sensor network implementation (which was not explicitly considered in [18]; trajectory-planning can be geared towards enhanced "pre-conditioning" of the partition-objective function as will be demonstrated for the target application.

It should be mentioned that a vast number of more conventional approaches exist which address MTT scenarios that are better suited for centralized implementation, see [20] for a fairly comprehensive overview of these methods. A large collection of algorithms have also recently been proposed that are based on Random Finite Set (RFS) theory [21] which use randomly varying sets to represent the target and measurement spaces. This formulation of the MTT problem can lead to a significantly more flexible representation of environments with an unknown number of targets and allows for occurrences such as sudden target appearance/disappearance, missing measurements and false alarms. Full incorporation of RFS theory in an MTT context has led to development of the "Multitarget Bayes Filter", that can be viewed as a generalization of the statistically optimal posterior filter in classical Bayesian theory. Unfortunately, the mathematical intractability that exists with the classical filter is dramatically amplified in the multitarget case and in many cases, even Monte Carlo techniques are prohibitively expensive in terms of computational resource necessary for sufficient approximation accuracy. Simplifications based on propagating moments of the multi-target distribution forward in time [22], as opposed to the distribution itself, have been developed which significantly reduce this cost; this reduction can be viewed as conceptually similar to Kalman filtering in a classical context, which propagates forward only the first two moments of the actual target posterior distribution. While RFS-theoretic based methods offer a promising direction in MTT research, they are outside of the main scope of this thesis. The focus in the current work is in further development and application of MASCOT to an environment with a fixed and known number of targets and where measurements are "globally influenced", i.e. one cannot directly associate any given measurement with a single target and therefore sensor misses/false alarms are not applicable here. As a result, application of an RFS-theoretic approach within the MASCOT framework would unnecessarily complicate matters and is withheld as an area for potential extension in the future.

Much of the work completed in [23] and [24] forms a basis for the proposed MASCOT system. In these papers, the concept of "switched" subspace-partitioning of the target state was outlined in the context of particle filtering. While similar to [18] at an abstract level, it is the belief here that [23] is a more effective implementation owing to a significantly higher-level of inter-agent collaboration. Specifically, for a given agent (filter), individual target predictions (each representing separately partitioned subspaces of the full target space) made at the previous time by other agents

are communicated to this agent and incorporated within computation of the weight-update equation that is necessary for the estimation of *its own* target state at the current time. This idea has been developed further within MASCOT; rather than conveying target estimates, probabilistic information is communicated by the other agents and properly fused into the marginal target posteriors. The general concept of this agent cooperation is illustrated in figure 2.1.

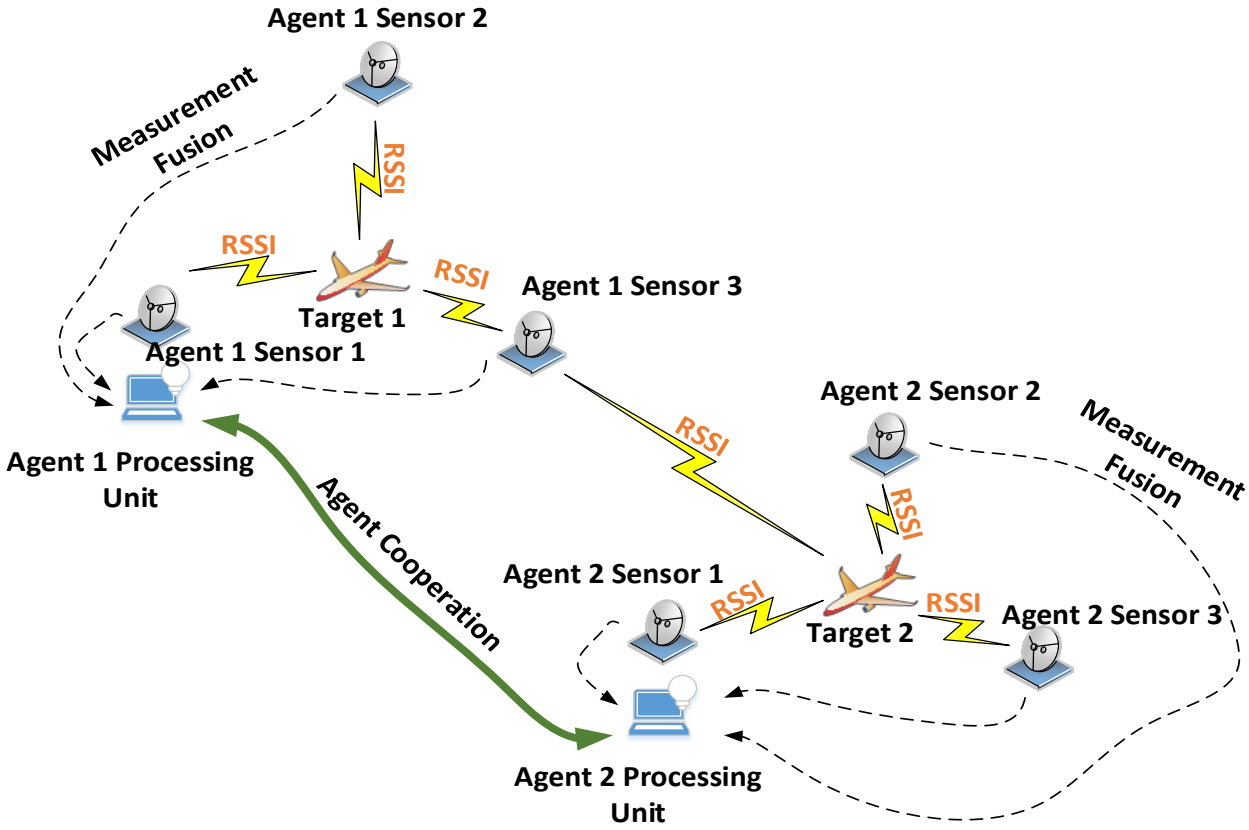


**Figure 2.1:** MASCOT Agent Cooperation

## 2.6 Detailed MASCOT Implementation in the MTT Environment

This subsection begins with a systematic definition of all notation required to describe MASCOT in the context of a MTT environment.

Consider the model as described in section 2.3 with a total of $L$ targets; the state of the $l$-th target at time $t$ is again described by the $4D$ vector $\mathbf{x}_{l,t} \in \mathbb{R}^4$. As in section 2.2 let us make a specific construction of the space $\mathbb{R}^{4L}$ that will be used to describe the state of all $L$ targets. Forming $L$ distinct copies of $\mathbb{R}^4$ labeled $\mathbb{R}^4_1, \mathbb{R}^4_2, \ldots, \mathbb{R}^4_L$, we have the construction,

$$\mathbb{R}^{4L} = \mathbb{R}^4_1 \times \mathbb{R}^4_2 \times \ldots \mathbb{R}^4_L = \prod_{l=1}^{L} \mathbb{R}^4_l \tag{2.28}$$

Notice the difference between (2.28) and (2.2); this time the space is constructed with multiples of $\mathbb{R}^4$ rather than $\mathbb{R}$ and is done since individual (scalar) coordinates corresponding to a particular target are always grouped together which greatly simplifies notation. The state of all targets at time $t$ is then described by the stacked vector in $\mathbb{R}^{4L}$ defined as $\mathbf{X}_t = \begin{bmatrix}\mathbf{x}_{1,t}^\top & \mathbf{x}_{2,t}^\top & \cdots & \mathbf{x}_{L,t}^\top\end{bmatrix}$. A total of $L$ agents is used to estimate the full target state; associated with the $i$-th agent at time $t$ are two indexing sets $\mathbf{S}_{i,t}$ and $\mathbf{F}_{i,t}$ defined as,

$$\mathbf{S}_{i,t} = \{j : j \in 1, 2, \ldots, L \text{ and } \mathbf{x}_{j,t} \text{ estimated by agent } i \text{ at time } t\} \tag{2.29}$$

$$\mathbf{F}_{i,t} = \mathbf{S}_{i,t}^c \tag{2.30}$$

Equipped with these sets, we can construct the subspaces $\mathbb{X}_{\mathbf{S}_{i,t}}$ and $\mathbb{X}_{\mathbf{F}_{i,t}}$ as,

$$\mathbb{X}_{\mathbf{S}_{i,t}} = \prod_{j \in \mathbf{S}_{i,t}} \mathbb{R}_j^4 \qquad \mathbb{X}_{\mathbf{F}_{i,t}} = \prod_{j \in \mathbf{F}_{i,t}} \mathbb{R}_j^4 \tag{2.31}$$

An important relation can also be stated here,

$$\mathbb{X}_{\mathbf{F}_{i,t}} = \prod_{j \neq i}^{L} \mathbb{X}_{\mathbf{S}_{j,t}} \tag{2.32}$$

Notice again that, in contrast to how these sets were defined in 2.2, whereby $j$ referred to a specific scalar component of the full state space, here $j$ refers to a specific target, which is instead a grouping of 4 scalar components in $\mathbf{X}_t$. Components of the state vector directly estimated by agent $i$, $\mathbf{X}_{\mathbf{S}_{i,t}} \in \mathbb{X}_{\mathbf{S}_{i,t}}$ can then be related back to the specific scalar components in $\mathbf{X}_t$ as,

$$\left[\mathbf{X}_{\mathbf{S}_{i,t}}\right]_{4(j-1)+1:4j} = [\mathbf{X}_t]_{4\left(\{\mathbf{S}_{i,t}\}_j - 1\right)+1:4\{\mathbf{S}_{i,t}\}_j} \qquad \text{for } j = 1 \ldots |\mathbf{S}_{i,t}| \tag{2.33}$$

The set $\mathbf{S}_{i,t}$ is labeled as the "in-focus" target index set of agent $i$ at time $t$, and its complement, $\mathbf{F}_{i,t}$ is labeled as the set of "foreign" target indices at time $t$ not estimated by agent $i$ and instead treated as measurement interference. It should be clear that the collection of subspaces $\mathbb{X}_{\mathbf{S}_{i,t}}$ for all agents, denoted by $\Theta_t$, represents a partition of the full target space, i.e,

$$\Theta_t = \left\{\mathbb{X}_{\mathbf{S}_{1,t}},\ \mathbb{X}_{\mathbf{S}_{2,t}},\ \cdots\ \mathbb{X}_{\mathbf{S}_{L,t}}\right\}, \tag{2.34}$$

$$\prod_{i=1}^{L} \mathbb{X}_{\mathbf{S}_{i,t}} = \mathbb{R}^{4L} \tag{2.35}$$

A fixed "pool" of $K$ sensors is used to generate observations regarding the MTT environment. Each sensor is dynamically assigned to one of the agents at time $t$ (Recall that agents treat their own sensor observations as private). While there are numerous possible ways to conduct sensor-to-agent assignment, the approach taken here is to *maintain a fixed and equal number of sensors allocated per target.* To be more precise, the fixed sensor pool is divided into $L$ groups, each containing $\frac{K}{L}$

sensors; each group is assigned to the same target. Let us then define $\mathbf{y}_{j,t}$ as the $\frac{K}{L}$ dimensional vector containing the $j$-th group of measurements. It is also sensible to maintain the sensor-target association over time, i.e., sensor group $j$ is associated with target $j$ for all $t$. As an example, let us assume $K = 4$ and $L = 2$. At time $t = 1$, agent 1 estimates target 1 only and is thus allocated sensors 1 and 2. If at time $t = 2$, the partition changes such that agent 1 is responsible for estimating both targets, it will be allocated all 4 sensors. With this assignment method in mind, the set $\mathbf{Y}_{i,t}$ associated with the $i$-th agent is defined as:

$$\mathbf{Y}_{i,t} = \{\mathbf{y}_{j,t} : j \in \mathbf{S}_{i,t}\} \tag{2.36}$$

Each sensor within the group producing the measurements $\mathbf{y}_{j,t}$ possesses a unique location; the vector $\mathbf{q}_{j,t}$ is used to vertically stack all locations within this group, i.e., the $2D$ component $[\mathbf{q}_{j,t}]_{\mathbf{k}}$ refers to the location of the $k$-th sensor within sensor group $j$ at time $t$. Note that an alternate notation which will not distinguish between particular groups within $\mathbf{Y}_{i,t}$ will also be used; namely $\{\mathbf{Y}_{i,t}\}_k$ will refer to simply the $k$-th measurement in this set, and $\{\mathbf{q}_{S_i,t}\}_k$ to its corresponding location.

The aim of agent $i$ at time $t$ is to represent the distribution $f(\mathbf{X}_{\mathbf{S}_{i,t}}|\mathbf{Y}_{i,1:t})$ with a random measure consisting of $M$ particles and denoted by $\chi_{i,t} = \{\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}, w^{(m)}_{\mathbf{S}_{i,t}}\}_{m=1}^{M}$. However, the dynamic partitioning of the state space must be kept in mind here; $\mathbf{X}_{\mathbf{S}_{i,t}}$ may not refer to the same state dimensions as $\mathbf{X}_{\mathbf{S}_{i,t-1}}$. For the moment, it is assumed that the target space partition remains the same when advancing from $t-1$ to $t$, i.e., $\Theta_t = \Theta_{t-1}$. Particles are then drawn as $\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}} \sim \pi_i(\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}|\mathbf{X}^{(m)}_{\mathbf{S}_{i,t-1}}, \mathbf{Y}_{i,t})$ and the weights are computed according to:

$$w^{(m)}_{\mathbf{S}_{i,t}} = w^{(m)}_{\mathbf{S}_{i,t-1}} \frac{f(\mathbf{Y}_{i,t}|\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}) f(\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}|\mathbf{X}^{(m)}_{\mathbf{S}_{i,t-1}})}{\pi_i(\mathbf{X}_{\mathbf{S}_{i,t}}|\mathbf{X}^{(m)}_{\mathbf{S}_{i,t-1}}, \mathbf{Y}_{i,t})} \tag{2.37}$$

The key obstacle here lies in the fact that the measurement set $\mathbf{Y}_{i,t}$ in general depends not only on $\mathbf{X}_{\mathbf{S}_{i,t}}$ but in fact the entire target state, $\mathbf{X}_t$. The step taken in [23] to get past this issue was to rewrite (2.37) as:

$$w^{(m)}_{\mathbf{S}_{i,t}} = w^{(m)}_{\mathbf{S}_{i,t-1}} \frac{f(\mathbf{Y}_{s_i,t}|\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}, \mathbf{X}_{\mathbf{F}_{i,t}}) f(\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}|\mathbf{X}^{(m)}_{\mathbf{S}_{i,t-1}}, \mathbf{X}_{\mathbf{F}_{i,t}})}{\pi_i(\mathbf{X}_{\mathbf{S}_{i,t}}|\mathbf{X}^{(m)}_{\mathbf{S}_{i,t-1}}, \mathbf{X}_{\mathbf{F}_{i,t-1}}, \mathbf{Y}_{i,t})} \tag{2.38}$$

and directly substitute a prediction made for $\mathbf{X}_{\mathbf{F}_{i,t}}$, labeled $\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}$ into (2.38), based on the estimates obtained for $\mathbf{X}_{\mathbf{F}_{i,t-1}}$ from the other agents. To better clarify this, the concept of inter-agent communication is now defined; this mechanism was present in [23] to a limited extent but plays a much greater role in the MASCOT framework. At each time $t$, agent $i$ receives information $\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)$ from all other agents (or possibly only those within a certain range) regarding its foreign states at the preceding time $t-1$. This information is fused into agent $i$'s estimation process in

order to form an estimate for $\mathbf{X}_{\mathbf{S}_{i,t}}$, labeled $\hat{\mathbf{X}}_{\mathbf{S}_{i,t}}$. In the case of [23], the information exchanged consisted of solely these estimates formed by the other agents, i.e., the vector $\hat{\mathbf{X}}_{\mathbf{F}_{i,t-1}} \in \mathbb{X}_{\mathbf{F}_{i,t-1}}$ formed by concatenating individual estimates from each of the other agents:

$$\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right) = \hat{\mathbf{X}}_{\mathbf{F}_{i,t-1}} = \begin{bmatrix} \mathbf{X}_{\mathbf{S}_{1,t-1}}^{\top} & \mathbf{X}_{\mathbf{S}_{2,t-1}}^{\top} & \dots & \mathbf{X}_{\mathbf{S}_{i-1,t-1}}^{\top} & \mathbf{X}_{\mathbf{S}_{i+1,t-1}}^{\top} \dots \mathbf{X}_{\mathbf{S}_{L,t-1}}^{\top} \end{bmatrix}^{\top} \tag{2.39}$$

Notice that (2.39) is consistent with the construction of $\mathbb{X}_{\mathbf{F}_{i,t-1}}$ according to (2.32). Equipped with this information, a prediction for $\mathbf{X}_{\mathbf{F}_{i,t}}$ can then be made by agent $i$ by propagating forward individual target estimates within $\hat{\mathbf{X}}_{\mathbf{F}_{i,t-1}}$:

$$\left[\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right]_{4(j-1)+1:4j} = \mathbf{A}\left[\hat{\mathbf{X}}_{\mathbf{F}_{i,t-1}}\right]_{4(j-1)+1:4j} \qquad \text{for } j = 1 \dots |\mathbf{F}_{i,t-1}| \tag{2.40}$$

While this method can be effective for situations where there is a small influence on the measurement set $\mathbf{Y}_{i,t}$ from $\mathbf{X}_{\mathbf{F}_{i,t}}$, the algorithm breaks down when this influence increases due to its inability to represent the uncertainty inherent in the foreign component estimates from other agents. As such, it is highly desirable to retain the form for the weight-updates in (2.37) and make approximations when possible to attain tractability.

This is indeed what is proposed here; to compute (2.37) with the modification that $f(\mathbf{Y}_{i,t}|\mathbf{X}_{\mathbf{S}_{i,t}}^{(m)})$ is approximated by $f\left(\mathbf{Y}_{i,t}|\mathbf{X}_{\mathbf{S}_{i,t}}^{(m)}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)$ for updating of the particle weights. It is observed that the $k$-th measurement in the set $\mathbf{Y}_{i,t}$ can be written as:

$$\{\mathbf{Y}_{i,t}\}_k = h_k\left(\mathbf{X}_{\mathbf{S}_{i,t}}, \mathbf{X}_{\mathbf{F}_{i,t}}\right) + v_{k,t} \tag{2.41}$$

where $v_{k,t}$ is the sensor noise and here assumed $\sim \mathcal{N}(0, \sigma_v^2)$. Thus, to obtain an expression for $f\left(\mathbf{Y}_{i,t}|\mathbf{X}_{\mathbf{S}_{i,t}}^{(m)}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)$, we can apply the method of transformation of random variables, given the known statistics of $v_{k,t}$ and the estimated statistics of $\mathbf{X}_{\mathbf{F}_{i,t}}$ derived from $\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)$. Specifically, the particle set representations of the posterior distributions of $\mathbf{X}_{\mathbf{F}_{i,t-1}}$ that are possessed by other agents within their in-focus partitions at time $t-1$ are used to formulate an approximation for the predictive distribution of $\mathbf{X}_{\mathbf{F}_{i,t}}$. This represents a core principle of MASCOT; that for a given agent, the probabilistic information, $\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)$, regarding $\mathbf{X}_{\mathbf{F}_{i,t-1}}$ is communicated by the other agents and used within the weight-update equation to form the agent's estimate of its own target partition. The specific information communicated can be in a number of different forms and can be considered as a form of "posterior handover" between the agents, see [25] for a detailed treatment of the subject. It has been chosen to communicate simple Gaussian approximations of the posteriors between agents; at time $t$, agent $i$ receives the information $\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)$ from other agents as:

$$\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right) = \{[\hat{\mathbf{x}}_{j,t-1}, \mathbf{C}_{j,t-1}] : j \in \mathbf{F}_{i,t-1}\} \tag{2.42}$$

To clarify further, for target $j$ which is contained within $\mathbf{X}_{\mathbf{F}_{i,t-1}}$, agent $i$ receives a mean and covariance estimate from some other agent (say agent $k$ such that $j \in \mathbf{S}_{k,t-1}$), agent $i$ then treats target $j$ as if $\mathbf{x}_{j,t-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{j,t-1}, \mathbf{C}_{j,t-1})$ when updating its particle weights at time $t$. Specifically, an estimate for the predictive distribution of $\mathbf{x}_{j,t}$ is first made by agent $i$ as:

$$\mathbf{x}_{j,t} \sim \mathcal{N}\Big(\mathbf{x}_{j,t} \mid \mathbf{A}\hat{\mathbf{x}}_{j,t-1}, \mathbf{A}\mathbf{C}_{j,t-1}\mathbf{A}^{\top} + \mathbf{Q}\Big) \tag{2.43}$$

This follows by referring back to the transition model in (2.10) and noting that if for some random vector $\mathbf{z} = \mathbf{A}\mathbf{u} + \mathbf{v}$ where $\mathbf{u} \sim \mathcal{N}(\hat{\mathbf{u}}, \mathbf{C}_u)$ and $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ then $\mathbf{z} \sim \mathcal{N}\left(\mathbf{A}\hat{\mathbf{u}}, \mathbf{A}\mathbf{C}_u\mathbf{A}^{\top} + \mathbf{Q}\right)$.

Let us then return to computation of the likelihood function $f(\mathbf{Y}_{i,t}|\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right))$ and determine how $\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)$ can be used. Referring back to the measurement model in (2.20) we can write:

$$\begin{aligned}
\{\mathbf{Y}_{i,t}\}_k &= h_k^{(s)}(\mathbf{X}_{\mathbf{S}_{i,t}}) + h_k^{(f)}(\mathbf{X}_{\mathbf{F}_{i,t}}) + v_{k,t} \\
h_k^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}\right) &= \sum_{j \in \mathbf{S}_{i,t}} \frac{\Phi}{\left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_k - \mathbf{x}_{j,t}^{1:2} \right\|^{\alpha} + \epsilon} \\
h_k^{(f)}\left(\mathbf{X}_{\mathbf{F}_{i,t}}\right) &= \sum_{j \in \mathbf{F}_{i,t}} \frac{\Phi}{\left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_k - \mathbf{x}_{j,t}^{1:2} \right\|^{\alpha} + \epsilon}
\end{aligned} \tag{2.44}$$

Where the $\mathbf{x}_{j,t}^{1:2}$ symbol is meant to convey that only the first two elements of this 4-D subvector (corresponding to the location coordinates) are taken. In computation of the likelihood, recall that the scalar $h_k^{(s)}(\mathbf{X}_{\mathbf{S}_{i,t}})$ is treated as deterministic, since we are conditioning on specific values of the particles $\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}$.

While in the most general multi-sensor case it is still believed that $f(\mathbf{Y}_{i,t}|\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right))$ is mathematically intractable, the author has found closed form expressions for specific cases where the sensor noise component $v_{k,t}$ is neglected, i.e., $f(\mathbf{Y}_{i,t} - v_{k,t}|\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right))$, and the measurement vector is either scalar or consists of two measurements. An approximation to the full likelihood can be computed by assuming sensor-independence (which is not technically correct but does produce reasonable results) and through convolution of this expression with the sensor noise pdf. Additionally, the closed form expressions obtained have allowed for development of a Gaussian approximation to the likelihood, which performs reasonably well under typical tracking conditions. While the details regarding these developments can be found in Chapter 6, the main result used here is that the likelihood can be approximated as follows:

$$\begin{aligned}
&f(\mathbf{Y}_{i,t}|\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)) \\
&\qquad \approx \prod_{j \in \mathbf{Y}_{i,t}} \mathcal{N}\left( \{\mathbf{Y}_{i,t}\}_j \mid h_j^{(s)}\left(\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}\right) + h_j^{(f)}\left(\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right), \left\{\sigma^{\star^2}\left(\mathbf{q}_{S_i,t}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j \right)
\end{aligned} \tag{2.45}$$

where $\mathcal{N}\left(x \mid \bar{x}, \sigma^2\right)$ is the pdf of a scalar Gaussian random variable with argument $x$, mean $\bar{x}$ and variance $\sigma^2$ and where,

$$h_j^{(f)}\left(\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right) = \sum_{k \in \mathbf{F}_{i,t}} \frac{\Phi}{\left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{k,t-1}\right)^{1:2} \right\|^{\alpha} + \epsilon} \tag{2.46}$$

$$\left\{\sigma^{\star^2}\left(\mathbf{q}_{S_i,t}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j = \sigma_v^2 + \sum_{k \in \mathbf{F}_{i,t}} \frac{\left\{\mathrm{D}^k_{\mathbf{S}_i,t}\right\}_j \Phi^2 \alpha^2 \left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{k,t-1}\right)^{1:2} \right\|^{2\alpha-4}}{\left( \left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{k,t-1}\right)^{1:2} \right\|^{\alpha} + \epsilon \right)^4} \tag{2.47}$$

$$\left\{\mathrm{D}^k_{\mathbf{S}_i,t}\right\}_j = \left(\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{k,t-1}\right)^{1:2}\right)^{\top} \left(\mathbf{A}\mathbf{C}_{k,t-1}\mathbf{A}^{\top} + \mathbf{Q}\right) \left(\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{k,t-1}\right)^{1:2}\right) \tag{2.48}$$

Equipped with this expression for $f(\mathbf{Y}_{i,t}|\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right))$, weights for the particles can then be properly computed as in (2.37), allowing agent $i$ to finally form its own information that will be transmitted to the other agents as:

$$\psi\left(\mathbf{X}_{\mathbf{S}_{i,t}}\right) = \left\{ [\hat{\mathbf{x}}_{j,t}, \mathbf{C}_{j,t}] = \left[ \sum_{m=1}^{M} w^{(m)}_{\mathbf{S}_{i,t}} \mathbf{x}^{(m)}_{j,t}, \sum_{m=1}^{M} w^{(m)}_{\mathbf{S}_{i,t}} \left(\mathbf{x}^{(m)}_{j,t} - \hat{\mathbf{x}}_{j,t}\right)\left(\mathbf{x}^{(m)}_{j,t} - \hat{\mathbf{x}}_{j,t}\right)^{\top} \right] : j \in \mathbf{S}_{i,t} \right\} \tag{2.49}$$

Notice that we can also write,

$$\hat{\mathbf{X}}_{\mathbf{S}_{i,t}} = \sum_{m=1}^{M} w^{(m)}_{\mathbf{S}_{i,t}} \mathbf{X}^{(m)}_{\mathbf{S}_{i,t}} \tag{2.50}$$

Equation (2.50) represents the final estimate produced by agent $i$ for the state $\mathbf{X}_{\mathbf{S}_{i,t}}$ and is embedded in the transmitted information $\psi\left(\mathbf{X}_{\mathbf{S}_{i,t}}\right)$. One could also form the covariance matrix $\mathbf{C}_{\mathbf{S}_{i,t}} = \sum_{m=1}^{M} w^{(m)}_{\mathbf{S}_{i,t}} \left(\hat{\mathbf{X}}_{\mathbf{S}_{i,t}} - \mathbf{X}_{\mathbf{S}_{i,t}}\right)\left(\hat{\mathbf{X}}_{\mathbf{S}_{i,t}} - \mathbf{X}_{\mathbf{S}_{i,t}}\right)^{\top}$ for the entire state $\mathbf{X}_{\mathbf{S}_{i,t}}$, however it was chosen in (2.50) to transmit only individual target covariance matrices since target motion is here modeled as independent.

Recall the preceding development assumed $\Theta_t = \Theta_{t-1}$. The case where the partition changes from $t-1$ to $t$ can be somewhat more complicated, however is alleviated to some extent by following the preceding notation as outlined. Several distinct cases of what may occur are outlined here with two simple scenarios; the full range of possible conditions can be covered by their straightforward extension. Let us first suppose that the in-focus space of agent $i$ is reduced in size by one target when advancing from $t-1$ to $t$, i.e., $\exists\, k$ s.t. $k \in \mathbf{S}_{i,t-1} \wedge k \notin \mathbf{S}_{i,t}$. Here, the particle filter algorithm can proceed at time $t$ as outlined previously with the minor adjustment that agent $i$ now must use the same information $[\hat{\mathbf{x}}_{k,t-1}, \mathbf{C}_{k,t-1}] \in \psi\left(\mathbf{X}_{\mathbf{S}_{i,t-1}}\right)$ it broadcast to other agents regarding the state of target $k$ at time $t-1$ to update its own weights at time $t$, i.e., this information *becomes part of* $\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)$. The second case considered consists of a target addition to $\mathbf{X}_{\mathbf{S}_{i,t}}$ from time $t-1$ to $t$, i.e., $\exists\ k$ s.t. $k \notin \mathbf{S}_{i,t-1} \wedge k \in \mathbf{S}_{i,t}$. Here the situation is somewhat more complex; the particle

representation $\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{(m)}, w_{\mathbf{S}_{i,t}}^{(m)}\right\}_{m=1:M}$ must grow in dimension by 4 to accommodate the additional target. There is also some ambiguity as to how to properly represent the joint distribution of $\mathbf{X}_{\mathbf{S}_{i,t}}$ so this is examined further here: we wish to represent $f\left(\mathbf{X}_{\mathbf{S}_{i,t}}|\mathbf{Y}_{i,1:t}\right)$ based on $\left\{\mathbf{X}_{\mathbf{S}_{i,t-1}}^{(m)}, w_{\mathbf{S}_{i,t}}^{(m)}\right\}_{m=1:M}$ and $[\hat{\mathbf{x}}_{k,t-1}, \mathbf{C}_{k,t-1}] \in \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)$. Since the underlying assumption is that the actual dependency between $\mathbf{X}_{\mathbf{S}_{i,t-1}}$ and $\mathbf{x}_{k,t-1}$ is relatively weak (else it would have been less likely that $k \notin \mathbf{S}_{i,t-1}$), *the product of the two distributions are taken for purposes of particle generation.* This can be done in a number of ways, but the approach taken here is to maintain a fixed particle size; assuming the transition prior is normally used for the proposal $\pi$, particle propagation at time $t$ is done by:

$$\mathbf{X}_{\mathbf{S}_{i,t}}^{(m)} = \mathbf{X}_{\mathbf{S}_{i,t}}^{(m)\prime} \cup \mathbf{x}_{k,t}^{(m)} \tag{2.51}$$

where the union is meant to convey that for each particle $\mathbf{X}_{\mathbf{S}_{i,t}}^{(m)}$ a *single* particle $\mathbf{x}_{k,t}^{(m)}$ is drawn and appended to the particle $\mathbf{X}_{\mathbf{S}_{i,t}}^{(m)\prime}$, where:

$$\begin{aligned}\mathbf{X}_{\mathbf{S}_{i,t}}^{(m)\prime} &= \left\{\mathbf{x}_{j,t}^{(m)} \sim \mathcal{N}\left(\mathbf{A}\mathbf{x}_{j,t-1}^{(m)}, \mathbf{Q}\right) : j \in \mathbf{S}_{i,t-1} \wedge j \in \mathbf{S}_{i,t}\right\} \\ \mathbf{x}_{k,t}^{(m)} &\sim \mathcal{N}\left(\mathbf{A}\hat{\mathbf{x}}_{k,t-1}, \mathbf{A}\mathbf{C}_{k,t-1}\mathbf{A}^{\top} + \mathbf{Q}\right)\end{aligned} \tag{2.52}$$

It should be clear that the weights can then be updated in a similar fashion as in the preceding examples.

As part of MASCOT, and outlined in a more general context within section 2.2, agents employ dynamic partitioning of the target state space according to (2.8). A sensible choice for the function $\mathbb{B}_i$ would be the trace of the Cramer-Rao-Lower-Bound (CRLB) matrix, $\mathbf{J}^{-1}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2} \mid \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)$, for all agents in a given *candidate partition* $\Theta_t'$. Note the symbol $\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}$ is used to emphasize that only the location coordinates of each target subvector within $\mathbf{X}_{\mathbf{S}_{i,t}}$ are taken. It can then be written[1] for each element of $\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}$:

$$\mathbb{E}\left[\left(\left\{\hat{\mathbf{X}}_{\mathbf{S}_{i,t}}^{1:2}\right\}_k - \left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_k\right)\left(\left\{\hat{\mathbf{X}}_{\mathbf{S}_{i,t}}^{1:2}\right\}_k - \left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_k\right)^{\top}\right] \succeq \left[\mathbf{J}^{-1}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2} \mid \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right]_{k,k} \tag{2.53}$$

where expectation is over the measurements $\mathbf{Y}_{i,t}$ only, and $\left\{\hat{\mathbf{X}}_{S_{i,t}}^{1:2}\right\}_k$ is any estimator of $\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_k$ that can be produced by agent $i$ given the externally communicated information $\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)$. Note the use of the subscript notation $\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_k$ here; this is meant to read as the $2 \times 1$ subvector in $\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}$ corresponding to the location coordinates of the $k$-th target within that vector, i.e.,

$$\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_k = \left[\mathbf{X}_{\mathbf{S}_{i,t}}\right]_{4(k-1)+1:4(k-1)+2} \tag{2.54}$$

Later the subscript $k^{(w)}$ will be used, as in $\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_{k^{(w)}}$ which is meant to refer to the $w$-th coordinate of the $k$-th subvector. Whenever the $k^{(w)}$ indexing notation is used to refer to a coordinate within a vector or matrix it is understood that targets and coordinates are always stacked

[1] For two matrices $\mathbf{A}$ and $\mathbf{B}$, the relation $\mathbf{A} \succeq \mathbf{B}$ means the matrix $\mathbf{A} - \mathbf{B}$ is positive semidefinite.

vertically. For example, if we take all vector elements from $\mathbf{X}^{1:2}_{\mathbf{S}_{i,t}}$ and stack them into a single vector, $\mathbf{U}$, of size $|\mathbf{X}^{1:2}_{\mathbf{S}_{i,t}}| \times 1$ ordered by coordinate and by target number, it is understood that $[\mathbf{U}]_{(2(k-1)+w)} = \left\{\mathbf{X}^{1:2}_{\mathbf{S}_{i,t}}\right\}_{k^{(w)}}$. To keep notation compact, the shorthand $[\mathbf{U}]_{k^{(w)}}$ is used to implicitly refer to $[\mathbf{U}]_{(2(k-1)+w)}$. With this in mind, we can see that $\mathbb{J}^{-1}\left(\mathbf{X}^{1:2}_{\mathbf{S}_{i,t}} \mid \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)$ is a $2|\mathbf{S}_{i,t}| \times 2|\mathbf{S}_{i,t}|$ matrix, and $\left[\mathbb{J}^{-1}\left(\mathbf{X}^{1:2}_{\mathbf{S}_{i,t}} \mid \psi\left(\mathbf{F}_{i,t-1}\right)\right)\right]_{k,k}$ refers to the $2 \times 2$ block on its diagonal.

$\mathbb{B}_i$ then acts as a lower bound on the best accuracy agent $i$ can achieve in estimating the target locations at time $t$ within its in-focus set given its current set of measurements. Let us now formulate an expression for $\mathbb{J}^{-1}\left(\mathbf{X}^{1:2}_{\mathbf{S}_{i,t}} \mid \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)$ assuming the external information $\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)$ is a Gaussian distribution as in (2.42):

$$\mathbf{J}^{-1}\left(\mathbf{X}^{1:2}_{\mathbf{S}_{i,t}} \mid \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) = \mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{i,t}}\mathbf{v}^{\top}_{\mathbf{S}_{i,t}}\right]^{-1}, \tag{2.55}$$

where,

$$\left[\mathbf{v}_{\mathbf{S}_{i,t}}\right]_{k^{(w)}} = \left(\frac{\partial \log f\left(\mathbf{Y}_{i,t} | \mathbf{X}^{1:2}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)}{\partial\left\{\mathbf{X}^{1:2}_{\mathbf{S}_{i,t}}\right\}_{k^{(w)}}}\right) \tag{2.56}$$

and expectation is taken over agent $i$'s set of measurements here, i.e.,

$$\mathbb{E}\left[\star\right] = \int (\star)\, f\left(\mathbf{Y}_{i,t} \mid \mathbf{X}^{1:2}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) d\mathbf{Y}_{i,t} \tag{2.57}$$

Referring back to (2.45), the log-likelihood is rewritten as:

$$\begin{aligned}\log f\left(\mathbf{Y}_{i,t} \mid \mathbf{X}^{1:2}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) &= \sum_{y_j \in \mathbf{Y}_{i,t}} \log f\left(y_j \mid \mathbf{X}^{1:2}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) \\ &\approx \sum_{y_j \in \mathbf{Y}_{i,t}} \log\left(\mathcal{N}\left(y_j \mid h^{(s)}_j\left(\mathbf{X}^{1:2}_{\mathbf{S}_{i,t}}\right) + h^{(f)}_j\left(\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right), \left\{\sigma^{\star 2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j\right)\right)\end{aligned} \tag{2.58}$$

Given that,

$$\mathcal{N}\left(x \mid \bar{x}, \sigma^2\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\bar{x})^2}{2\sigma^2}\right] \tag{2.59}$$

We can express (2.58) as:

$$\begin{aligned}\log f\left(\mathbf{Y}_{i,t} \mid \mathbf{X}^{1:2}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) \approx \sum_{y_j \in \mathbf{Y}_{i,t}} \Bigg[ &\frac{-\left(y_j - \left[h^{(s)}_j\left(\mathbf{X}^{1:2}_{\mathbf{S}_{i,t}}\right) + h^{(f)}_j\left(\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right)\right]\right)^2}{2\left\{\sigma^{\star 2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j} \\ &- \log\left(\sqrt{2\pi}\left\{\sigma^{\star}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j\right)\Bigg]\end{aligned} \tag{2.60}$$

Taking the derivative of this expression with respect to $\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_{k^{(w)}}$, we have:

$$\frac{\partial \log f\left(\mathbf{Y}_{i,t} \mid \mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)}{\partial\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_{k^{(w)}}} \approx \sum_{y_j \in \mathbf{Y}_{i,t}} \left(\frac{\left(y_j - \left[h_j^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right) + h_j^{(f)}\left(\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right)\right]\right)}{\left\{\sigma^{\star 2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j}\right)\left(\frac{\partial h_j^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right)}{\partial\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_{k^{(w)}}}\right) \tag{2.61}$$

The derivative term is evaluated using (2.44):

$$\begin{aligned}\frac{\partial h_j^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right)}{\partial\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_{k^{(w)}}} &= \frac{\partial}{\partial\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_{k^{(w)}}}\left(\sum_{k' \in \mathbf{S}_{i,t}} \frac{\Phi}{\left\|\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \mathbf{x}_{k',t}^{1:2}\right\|^{\alpha} + \epsilon}\right) \\ &= \frac{\alpha\Phi\left(\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_{j^{(w)}} - \mathbf{x}_{k,t}^{(w)}\right)\left\|\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \mathbf{x}_{k,t}^{1:2}\right\|^{\alpha-2}}{\left(\left\|\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \mathbf{x}_{k,t}^{1:2}\right\|^{\alpha} + \epsilon\right)^2}\end{aligned} \tag{2.62}$$

Referring back to (2.55), we then can write:

$$\begin{aligned}&\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{i,t}}\mathbf{v}_{\mathbf{S}_{i,t}}^{\top}\right]_{k^{(w)},l^{(z)}} = \\ &\quad\mathbb{E}\left[\left(\sum_{y_j \in \mathbf{Y}_{i,t}}\left(\frac{\left(y_j - \left[h_j^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right) + h_j^{(f)}\left(\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right)\right]\right)}{\left\{\sigma^{\star 2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j}\right)\left(\frac{\partial h_j^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right)}{\partial\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_{k^{(w)}}}\right)\right)\right. \\ &\quad\left.\times\left(\sum_{y_j \in \mathbf{Y}_{i,t}}\left(\frac{\left(y_j - \left[h_j^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right) + h_j^{(f)}\left(\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right)\right]\right)}{\left\{\sigma^{\star 2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j}\right)\left(\frac{\partial h_j^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right)}{\partial\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_{l^{(z)}}}\right)\right)\right] \\ &\quad= \sum_{\substack{y_j \in \mathbf{Y}_{i,t} \\ y_p \in \mathbf{Y}_{i,t}}}\left(\frac{\left(\frac{\partial h_j^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right)}{\partial\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_{k^{(w)}}}\right)\left(\frac{\partial h_p^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right)}{\partial\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_{l^{(z)}}}\right)}{\left\{\sigma^{\star}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j^2\left\{\sigma^{\star}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_p^2}\right. \\ &\quad\left.\times\,\mathbb{E}\left[\left(y_j - h_j^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right) - h_j^{(f)}\left(\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right)\right)\left(y_p - h_p^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right) - h_p^{(f)}\left(\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right)\right)\right]\right)\end{aligned} \tag{2.63}$$

Referring to (2.46) and noting that:

$$\mathbb{E}\left[\left(y_j - h_j^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right) - h_j^{(f)}\left(\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right)\right)\left(y_p - h_p^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right) - h_p^{(f)}\left(\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right)\right)\right] = \begin{cases}\left\{\sigma^{\star}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j^2, & \text{if } j = p. \\ 0, & \text{otherwise.}\end{cases} \tag{2.64}$$

We have finally that:

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{i,t}}\mathbf{v}_{\mathbf{S}_{i,t}}^{\top}\right]_{k^{(w)},l^{(z)}} = \sum_{y_j\in\mathbf{Y}_{i,t}} \frac{\left(\frac{\partial h_j^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right)}{\partial\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_{k^{(w)}}}\right)\left(\frac{\partial h_j^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right)}{\partial\left\{\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right\}_{l^{(z)}}}\right)}{\left\{\sigma^{\star}\left(\mathbf{q}_{\mathbf{S}_{i,t}},\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j^2}$$

$$= \sum_{y_j\in\mathbf{Y}_{i,t}} \frac{\alpha^2\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_{j^{(w)}} - \mathbf{x}_{k,t}^{(w)}\right)\left(\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_{j^{(z)}} - \mathbf{x}_{l,t}^{(z)}\right)\left(\left\|\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \mathbf{x}_{k,t}^{1:2}\right\|\left\|\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \mathbf{x}_{l,t}^{1:2}\right\|\right)^{\alpha-2}}{\left(\left\{\sigma^{\star}\left(\mathbf{q}_{\mathbf{S}_{i,t}},\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j\left(\left\|\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \mathbf{x}_{k,t}^{1:2}\right\|^{\alpha} + \epsilon\right)\left(\left\|\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \mathbf{x}_{l,t}^{1:2}\right\|^{\alpha} + \epsilon\right)\right)^2} \tag{2.65}$$

The matrix formed based on the preceding expression for each of the elements must then be inverted to yield the CRLB matrix, $\mathbf{J}^{-1}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2} \mid \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)$ as noted in (2.55).

Let us now consider a possible expression for $\mathbb{D}_M\left(\mathbb{X}_{\mathbf{S}_{i,t}},\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)$ which can be considered as a regularizing function that allows MASCOT to favor partitions with lowest dimensionality. This function is of great significance, since in the absence of it, the function $\mathbb{B}_i$ will generally favor partitions with higher dimensionality. While the choice of an error-bound for $\mathbb{B}_i$ is reasonable, it must be understood that it provides only a benchmark for the best-achievable accuracy any estimator can attain; it provides no information regarding how well a specific suboptimal estimator, i.e., the particle filter, will do and can be misleading. Ideally, the function $\mathbb{D}_M$ should be naturally derived based on the estimation algorithm chosen, i.e., a prediction of how well the particle filter will perform given the value of the eror bound, number of particles, and dimensionality of the system. Work done in [26],[27],[28], [29], and many others offers insight into how adequate particle sampling coverage is affected by the dimensionality of the problem. Unfortunately, there does not seem to be a precise, tractable, formula for estimating how "far from optimal" a given particle filter will be. One possibility involves leveraging the result in [29] which shows for the particular problem considered in that paper, the Monte-Carlo error grows at the rate of $O\left(\frac{e^{\frac{\alpha}{4}d}}{\sqrt{N}}\right)$ where $\alpha \geq 2\min(T,1)$ and $T$ is the time interval unit. While the problem considered here is not entirely similar to that in [29], this expression indeed has been found empirically to work well as $\mathbb{D}_M$. To be more precise, it is proposed here to set $D_M\left(\mathbb{X}_{\mathbf{S}_{i,t}},\psi\left(\mathbf{X}_{\mathbf{F}_{i,t}}\right)\right) = e^{|\mathbf{S}_{i,t}|}$; notice here the assumption [1] that $\alpha = 2$, and $d = 2|\mathbf{S}_{i,t}|$. A rigorous derivation of $D_M$ that should also depend on the number of particles and properties of the observation likelihood is left as an open area for future work.

Finally, equipped with an expression for $\mathbb{B}_i$ and $\mathbb{D}_M$, the objective for a given candidate partition $\Theta_t'$ is then computed as in (2.8). The candidate partition, $\Theta_t^{\star}$ yielding the lowest value of $\mathcal{O}_t$ is chosen to represent $\mathbf{X}_t$ at time $t$, i.e., $\Theta_t = \Theta_t^{\star}$.

To illustrate how MASCOT works, let us consider an example consisting of two targets and two

[1] Although d is actually $4|\mathbf{S}_{i,t}|$, the velocity terms are conditionally linear and Rao-Blackwellization can be applied to reduce the effective dimension to 2

agents ($L = 2$), with a pool of $K = 4$ sensors, and for simplicity $\epsilon = 0, \alpha = 2$. For this example, let us set $\mathbb{D}_M\left(\mathbb{X}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) = 1$ and $(\lambda_1, \lambda_2) = (0, 0)$ corresponding to a desire for maximum estimation accuracy without regard for dimensionality. At each time $t$, there are only *two* possible unique partitions of $\mathbf{X}_t$:

$$\begin{aligned}\Theta_t^{(1)} &= \left\{\mathbb{R}_1^4 \times \mathbb{R}_2^4\right\} \\ \Theta_t^{(2)} &= \left\{\mathbb{R}_1^4, \mathbb{R}_2^4\right\}\end{aligned} \tag{2.66}$$

The candidate partition $\Theta_t^{(1)}$ consists of a single agent estimating both targets , i.e., $\mathbf{X}_{\mathbf{S}_{1,t}} = \mathbf{X}_t$ and $\mathbf{X}_{\mathbf{S}_{2,t}} = \emptyset$ while the candidate partition $\Theta_t^{(2)}$ assigns a single target to each agent, i.e., $\mathbf{X}_{\mathbf{S}_{1,t}} = \mathbf{x}_{1,t}$ and $\mathbf{X}_{\mathbf{S}_{2,t}} = \mathbf{x}_{2,t}$. The quantity $\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^\top\right]$ under candidate partition $\Theta_t^{(1)}$ (labeled $\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^\top\right]^{\Theta_t^{(1)}}$ is then a $4 \times 4$ matrix with the elements:

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^\top\right]_{1,1}^{\Theta_t^{(1)}} = \sum_{y_j \in \mathbf{Y}_{1,t}} \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(1)}} - \mathbf{x}_{1,t}^{(1)}\right)^2}{\sigma_v^2\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{1,t}^{1:2}\right\|^8}$$

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^\top\right]_{1,2}^{\Theta_t^{(1)}} = \mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^\top\right]_{2,1}^{\Theta_t^{(1)}} = \sum_{y_j \in \mathbf{Y}_{1,t}} \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(1)}} - \mathbf{x}_{1,t}^{(1)}\right)\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(2)}} - \mathbf{x}_{1,t}^{(2)}\right)}{\sigma_v^2\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{1,t}^{1:2}\right\|^8}$$

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^\top\right]_{2,2}^{\Theta_t^{(1)}} = \sum_{y_j \in \mathbf{Y}_{1,t}} \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(2)}} - \mathbf{x}_{1,t}^{(2)}\right)^2}{\sigma_v^2\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{1,t}^{1:2}\right\|^8}$$

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^\top\right]_{1,3}^{\Theta_t^{(1)}} = \mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^\top\right]_{3,1}^{\Theta_t^{(1)}} = \sum_{y_j \in \mathbf{Y}_{1,t}} \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(1)}} - \mathbf{x}_{1,t}^{(1)}\right)\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(1)}} - \mathbf{x}_{2,t}^{(1)}\right)}{\sigma_v^2\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{1,t}^{1:2}\right\|^4\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{2,t}^{1:2}\right\|^4}$$

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^\top\right]_{1,4}^{\Theta_t^{(1)}} = \mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^\top\right]_{4,1}^{\Theta_t^{(1)}} = \sum_{y_j \in \mathbf{Y}_{1,t}} \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(1)}} - \mathbf{x}_{1,t}^{(1)}\right)\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(2)}} - \mathbf{x}_{2,t}^{(2)}\right)}{\sigma_v^2\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{1,t}^{1:2}\right\|^4\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{2,t}^{1:2}\right\|^4}$$

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^{\top}\right]_{2,3}^{\Theta_t^{(1)}} = \mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^{\top}\right]_{3,2}^{\Theta_t^{(1)}} = \sum_{y_j\in\mathbf{Y}_{1,t}} \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(2)}} - \mathbf{x}_{1,t}^{(2)}\right)\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(1)}} - \mathbf{x}_{2,t}^{(1)}\right)}{\sigma_v^2\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{1,t}^{1:2}\right\|^4\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{2,t}^{1:2}\right\|^4}$$

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^{\top}\right]_{2,4}^{\Theta_t^{(1)}} = \mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^{\top}\right]_{4,2}^{\Theta_t^{(1)}} = \sum_{y_j\in\mathbf{Y}_{1,t}} \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(2)}} - \mathbf{x}_{1,t}^{(2)}\right)\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(2)}} - \mathbf{x}_{2,t}^{(2)}\right)}{\sigma_v^2\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{1,t}^{1:2}\right\|^4\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{2,t}^{1:2}\right\|^4}$$

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^{\top}\right]_{3,4}^{\Theta_t^{(1)}} = \mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^{\top}\right]_{4,3}^{\Theta_t^{(1)}} = \sum_{y_j\in\mathbf{Y}_{1,t}} \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(1)}} - \mathbf{x}_{2,t}^{(1)}\right)\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(2)}} - \mathbf{x}_{2,t}^{(2)}\right)}{\sigma_v^2\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{2,t}^{1:2}\right\|^8}$$

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^{\top}\right]_{3,3}^{\Theta_t^{(1)}} = \sum_{y_j\in\mathbf{Y}_{1,t}} \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(1)}} - \mathbf{x}_{2,t}^{(1)}\right)^2}{\sigma_v^2\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{2,t}^{1:2}\right\|^8}$$

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^{\top}\right]_{4,4}^{\Theta_t^{(1)}} = \sum_{y_j\in\mathbf{Y}_{1,t}} \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(2)}} - \mathbf{x}_{2,t}^{(2)}\right)^2}{\sigma_v^2\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{2,t}^{1:2}\right\|^8} \tag{2.67}$$

where it is noted that $\left\{\sigma^\star\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j = \sigma_v \; \forall \, j$ since $\mathbf{X}_{\mathbf{F}_{1,t}} = \emptyset$[1] Moving on to candidate partition $\Theta_t^{(2)}$, it is noted that the quantities $\mathbb{E}\left[\mathbf{v}_{S_1,t}\mathbf{v}_{S_1,t}^{\top}\right]^{\Theta_t^{(2)}}$ and $\mathbb{E}\left[\mathbf{v}_{S_2,t}\mathbf{v}_{S_2,t}^{\top}\right]^{\Theta_t^{(2)}}$ are both $2\times 2$ matrices with elements:

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^{\top}\right]_{w,z}^{\Theta_t^{(2)}} =$$

$$\sum_{y_j\in\mathbf{Y}_{1,t}} \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(w)}} - \mathbf{x}_{1,t}^{(w)}\right)\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_{j^{(z)}} - \mathbf{x}_{1,t}^{(z)}\right)}{\left(\sigma_v^2 + \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{2,t-1}\right)^{1:2}\right)^{\top}\left(\mathbf{A}\mathbf{C}_{2,t-1}\mathbf{A}^{\top}+\mathbf{Q}\right)\left(\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{2,t-1}\right)^{1:2}\right)}{\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{2,t-1}\right)^{1:2}\right\|^8}\right)\left\|\left\{\mathbf{q}_{\mathbf{S}_{1,t}}\right\}_j - \mathbf{x}_{1,t}^{1:2}\right\|^8}$$

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{2,t}}\mathbf{v}_{\mathbf{S}_{2,t}}^{\top}\right]_{w,z}^{\Theta_t^{(2)}} =$$

$$\sum_{y_j\in\mathbf{Y}_{2,t}} \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{2,t}}\right\}_{j^{(w)}} - \mathbf{x}_{2,t}^{(w)}\right)\left(\left\{\mathbf{q}_{\mathbf{S}_{2,t}}\right\}_{j^{(z)}} - \mathbf{x}_{2,t}^{(z)}\right)}{\left(\sigma_v^2 + \frac{4\Phi^2\left(\left\{\mathbf{q}_{\mathbf{S}_{2,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{1,t-1}\right)^{1:2}\right)^{\top}\left(\mathbf{A}\mathbf{C}_{1,t-1}\mathbf{A}^{\top}+\mathbf{Q}\right)\left(\left\{\mathbf{q}_{\mathbf{S}_{2,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{1,t-1}\right)^{1:2}\right)}{\left\|\left\{\mathbf{q}_{\mathbf{S}_{2,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{1,t-1}\right)^{1:2}\right\|^8}\right)\left\|\left\{\mathbf{q}_{\mathbf{S}_{2,t}}\right\}_j - \mathbf{x}_{2,t}^{1:2}\right\|^8} \tag{2.68}$$

Notice that in both equations (2.67) and (2.68), the in-focus targets of each partition ($\mathbf{x}_{1,t}, \mathbf{x}_{2,t} \in \mathbb{X}_{\mathbf{S}_{1,t}}$) for $\Theta_t^{(1)}$ and $\mathbf{x}_{1,t} \in \mathbb{X}_{\mathbf{S}_{1,t}}, \mathbf{x}_{2,t} \in \mathbb{X}_{\mathbf{S}_{2,t}}$ for $\Theta_t^{(2)}$ are unknown at time $t$; these quantities are eval-

[1] Note that $\mathbb{B}_i = 0$ if $\mathbf{X}_{\mathbf{S}_{i,t}} = \emptyset$.

uated at predictions of each target, i.e., for $\Theta_t^{(2)}$, $\mathbf{x}_{1,t}$ is replaced with $\mathbf{A}\hat{\mathbf{x}}_{1,t-1}$ in $\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}^\top\right]_{w,z}^{\Theta_t^{(2)}}$. The substitution was intentionally omitted from the preceding equations to differentiate between targets from $\mathbf{X}_{\mathbf{S}_{i,t}}$ and those from $\mathbf{X}_{\mathbf{F}_{i,t}}$.

It is clear that each of the objective functions, $\mathbb{B}_i(\Theta_t)$, have a strong dependence on the relative offsets between the sensors and targets; when evaluating candidate partition objectives, it is sensible to compute these values assuming sensors will be arranged in an optimal configuration for the corresponding partition. Optimal sensor positioning is for the moment considered a separate issue and is addressed in section 2.7. For now, let us assume each sensor pair is positioned along a circle of fixed radius $r$ about one of the targets (this is an optimal configuration for a single target scenario with two sensors but suboptimal here). As discussed previously and noted by (2.36), sensors are allocated to each agent in such a way that the sensor-target association remains fixed over time. For this example, let us denote the $k$-th pair of sensor measurements by $\mathbf{y}_{k,t}$, then we have the following sensor allocation for each candidate partition:

$$\begin{aligned} \mathbf{Y}_{1,t}^{\Theta_t^{(1)}} &= \{\mathbf{y}_{1,t}, \mathbf{y}_{2,t}\} \qquad \mathbf{Y}_{2,t}^{\Theta_t^{(1)}} = \emptyset \\ \mathbf{Y}_{1,t}^{\Theta_t^{(2)}} &= \{\mathbf{y}_{1,t}\} \qquad \mathbf{Y}_{2,t}^{\Theta_t^{(2)}} = \{\mathbf{y}_{2,t}\} \end{aligned} \tag{2.69}$$

To clarify further, if at time $t$, the partition $\Theta_t^{(1)}$ is chosen, then sensors are allocated according to $\mathbf{Y}_{i,t}^{\Theta_t^{(1)}}$, otherwise if $\Theta_t^{(2)}$ is chosen, sensors are allocated according to $\mathbf{Y}_{i,t}^{\Theta_t^{(2)}}$.

While the partitioning objective function has a relatively complex form even in the case of this simple two-target configuration, let us consider a specific scenario to see if any informative observations can be made. Let us assume that the two targets both lie along the horizontal axis and are initially separated by a distance $2d \geq 2r$, i.e., $\mathbf{x}_{1,t} = \begin{bmatrix} d & 0 \end{bmatrix}^\top$ and $\mathbf{x}_{2,t} = \begin{bmatrix} -d & 0 \end{bmatrix}^\top$. Sensor positioning is done in both candidate partitions according to:

$$\begin{aligned} [\mathbf{q}_{1,t}]_1 &= \mathbf{x}_{1,t} + \begin{bmatrix} \frac{r}{\sqrt{2}} & \frac{r}{\sqrt{2}} \end{bmatrix}^\top \qquad [\mathbf{q}_{1,t}]_2 = \mathbf{x}_{1,t} + \begin{bmatrix} \frac{r}{\sqrt{2}} & -\frac{r}{\sqrt{2}} \end{bmatrix}^\top \\ [\mathbf{q}_{2,t}]_1 &= \mathbf{x}_{2,t} - \begin{bmatrix} \frac{r}{\sqrt{2}} & -\frac{r}{\sqrt{2}} \end{bmatrix}^\top \qquad [\mathbf{q}_{2,t}]_2 = \mathbf{x}_{2,t} - \begin{bmatrix} \frac{r}{\sqrt{2}} & -\frac{r}{\sqrt{2}} \end{bmatrix}^\top \end{aligned} \tag{2.70}$$

In positioning the sensors and in subsequent evaluation of the candidate objectives, let us assume each agent makes "perfect" predictions, i.e., $\hat{\mathbf{x}}_{j,t} = \mathbf{x}_{j,t}$ but also predicts the covariance matrix of each target $\in \mathbf{X}_{\mathbf{F}_{i,t}}$, $\mathbf{C}_{j,t} = \mathbf{A}\mathbf{C}_{j,t-1}\mathbf{A} + \mathbf{Q}$ to be $\sigma_f^2\mathbf{I}$. It can be shown that the matrices $\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{i,t}}\mathbf{v}_{\mathbf{S}_{i,t}}\right]$

take on the form:

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}\right]^{\Theta_t^{(1)}} =$$

$$\frac{4\Phi^2}{\sigma_v^2}\begin{bmatrix} \frac{1}{r^6}+\frac{\left(4d+\sqrt{2}r\right)^2}{2\left(4d^2+2\sqrt{2}dr+r^2\right)^4} & 0 & \frac{2\left(2\sqrt{2}d+r\right)}{r^3\left(4d^2+2\sqrt{2}dr+r^2\right)^2} & 0 \\ 0 & \frac{1}{r^6}+\frac{r^2}{\left(4d^2+2\sqrt{2}dr+r^2\right)^4} & 0 & \frac{2}{r^2\left(4d^2+2\sqrt{2}dr+r^2\right)^2} \\ \frac{2\left(2\sqrt{2}d+r\right)}{r^3\left(4d^2+2\sqrt{2}dr+r^2\right)^2} & 0 & \frac{1}{r^6}+\frac{\left(4d+\sqrt{2}r\right)^2}{2\left(4d^2+2\sqrt{2}dr+r^2\right)^4} & 0 \\ 0 & \frac{2}{r^2\left(4d^2+2\sqrt{2}dr+r^2\right)^2} & 0 & \frac{1}{r^6}+\frac{r^2}{\left(4d^2+2\sqrt{2}dr+r^2\right)^4} \end{bmatrix} \tag{2.71}$$

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}\right]^{\Theta_t^{(2)}} = \mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{2,t}}\mathbf{v}_{\mathbf{S}_{2,t}}\right]^{\Theta_t^{(2)}} = \left(\frac{4\Phi^2}{r^6\left(\sigma_v^2+\frac{4\Phi^2\sigma_f^2}{\left(\frac{r^2}{2}+\left(2d+\frac{r}{\sqrt{2}}\right)^2\right)^3}\right)}\right)\mathbf{I}_2 \tag{2.72}$$

To find the objective functions for each candidate partition, $\mathcal{O}_t^{\Theta_t^{(l)}}$, these matrices must be inverted and the matrix trace operator is then applied. This yields a simple form for $\Theta_t^{(2)}$ in this scenario, as both its matrices in (2.72) are diagonal,

$$\mathbb{B}_t\left(\Theta_t^{(2)}\right) = \frac{r^6}{\Phi^2}\left(\sigma_v^2+\frac{4\Phi^2\sigma_f^2}{\left(\frac{r^2}{2}+\left(2d+\frac{r}{\sqrt{2}}\right)^2\right)^3}\right) \tag{2.73}$$

The objective function for $\Theta_t^{(1)}$ is considerably more complex owing to the necessary $4\times 4$ matrix inversion and its final form is not shown (it is a relatively high order polynomial in $d$ and $r$).
One clear observation that can be made here is that $\mathbb{B}_i \to \frac{r^6\sigma_v^2}{\Phi^2}$ as $d\to\infty$ in either candidate. This can be seen easily in the case of $\Theta_t^{(2)}$ since the term involving $\sigma_f^2$ in (2.73) decays rapidly as $d$ increases. While not so obvious for $\Theta_t^{(1)}$, it is straightforward to show that all terms involving $d$ in (2.71) decay to 0, leaving us with $\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{1,t}}\mathbf{v}_{\mathbf{S}_{1,t}}\right]^{\Theta_t^{(1)}} \to \frac{4\Phi^2}{\sigma_v^2 r^6}\mathbf{I}_4$ as $d\to\infty$, which clearly produces the same result upon taking the trace of its inverse. The candidate that should be chosen by MASCOT (with a lower value for $\mathcal{O}_t$) is difficult to quantify even in this simple scenario due to the complex form of $\mathbb{B}_i\left(\Theta_t^{(1)}\right)$ and is significantly influenced by the various parameters $(\Phi,\sigma_v,\sigma_f,r)$. Clearly, as the target separation increases, choosing $\Theta_t^{(2)}$ will be more favorable since, as shown, the two candidate objectives converge to the same value, thus there is no longer any benefit in choosing $\Theta_t^{(1)}$ which would yield a higher-dimensional partition and cause a larger value for $\mathbb{D}_M$ in $\mathcal{O}_t$. However,

the situation can be significantly different when the targets are in close proximity. Figures 2.2 and 2.3 illustrate this variation in $\mathbb{B}_i(\Theta_t)$ for different values of $\Phi$. Notice how as $\Phi$ increases, the candidate $\Theta_t^{(1)}$ yields increasingly better performance over $\Theta_t^{(2)}$ for smaller target separations; this is intuitive since at higher power levels, there will be a larger amount of interference in sensor measurements from the foreign target.

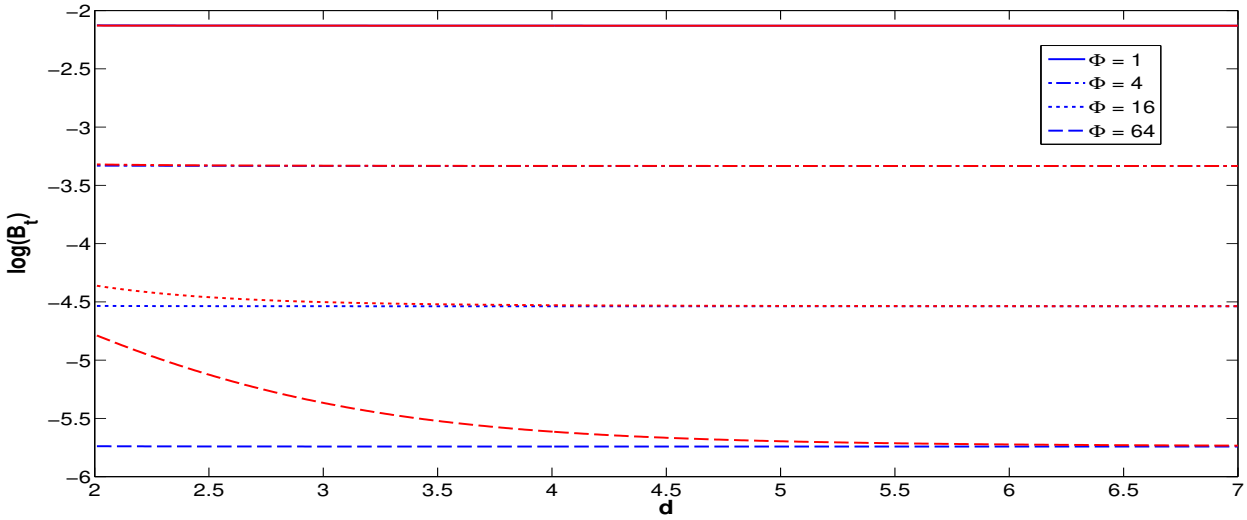


**Figure 2.2:** Plot of $\log \mathbb{B}_t(\Theta_t)$ for $(\sigma_v, \sigma_f) = (0.01, 0.04)$ for various values of $\Phi$ and $r = 2$. Blue lines indicate $\Theta_t^{(1)}$, and red indicate $\Theta_t^{(2)}$.

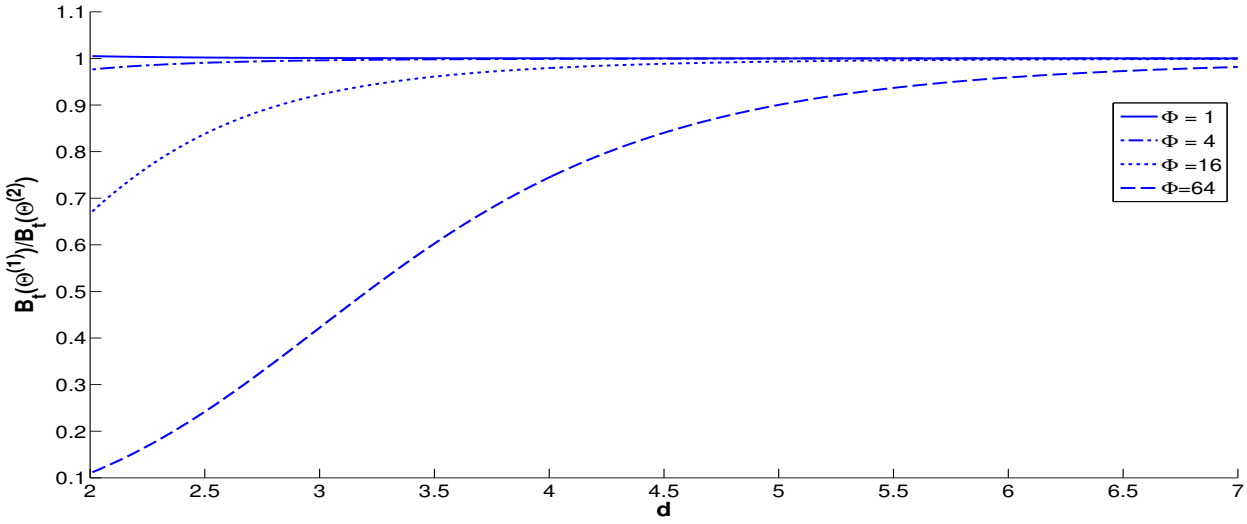


**Figure 2.3:** Plot of the ratio $\frac{\mathbb{B}_i\left(\Theta_t^{(1)}\right)}{\mathbb{B}_i\left(\Theta_t^{(2)}\right)}$ for $(\sigma_v, \sigma_f) = (0.01, 0.04)$ for various values of $\Phi$ and $r = 2$.

Figure 2.4 examines the ratio $\frac{\mathbb{B}_i\left(\Theta_t^{(1)}\right)}{\mathbb{B}_i\left(\Theta_t^{(2)}\right)}$ for a range of $(\sigma_f, r)$ at the fixed target separation $d = 2$

and fixed power $\Phi = 16$. Notice that a green plane for $\frac{\mathbb{B}_i\left(\Theta_t^{(1)}\right)}{\mathbb{B}_i\left(\Theta_t^{(2)}\right)} = 1$ separates locations where $\Theta_t^{(2)}$ is favorable (above) and where $\Theta_t^{(1)}$ is favorable (below). It can be seen that higher values of $\sigma_f$ result in $\Theta_t^{(1)}$ to be favored; this can be expected since there is less certainty in the exchanged information between the two agents (for $\Theta_t^{(2)}$) resulting in a higher level of interference to sensor measurements. A similar explanation can be stated for decreasing values of $r$; if the sensors are closer to their "target of interest", they will also be closer to the interfering target if there is small target-separation, resulting in a higher performance degradation.

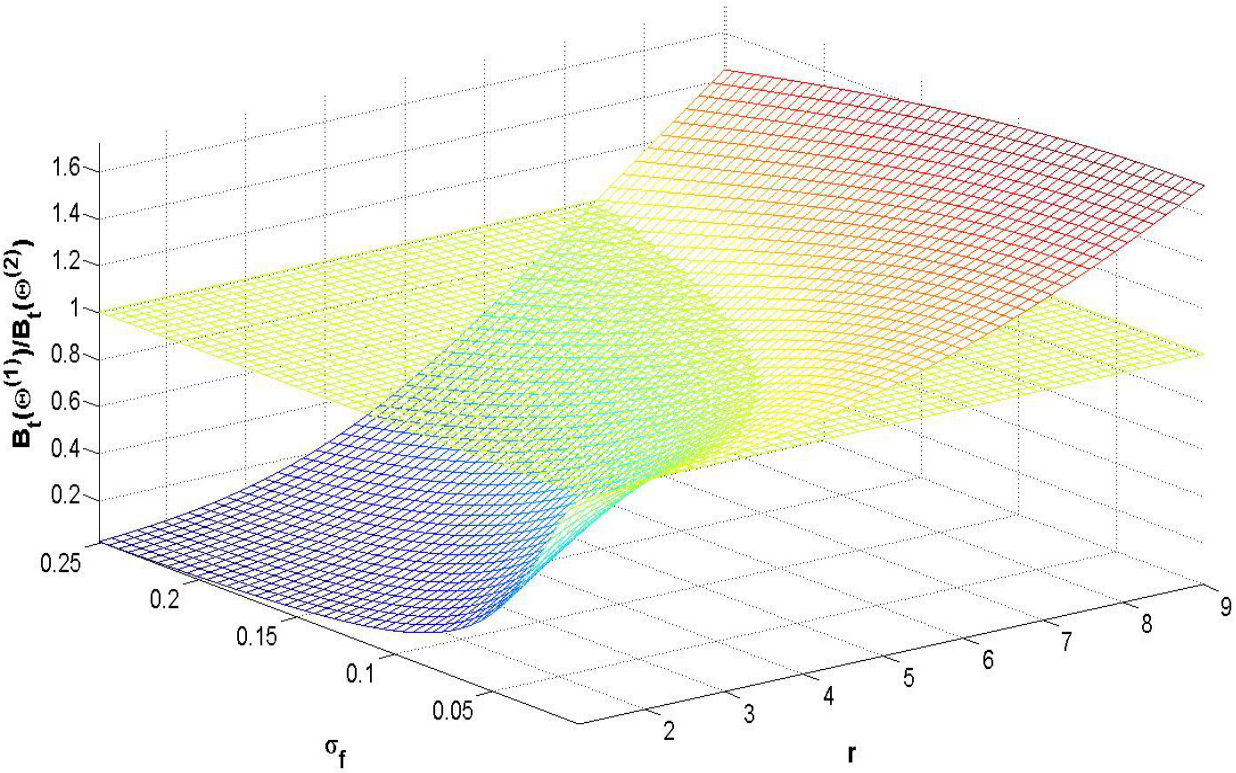


**Figure 2.4:** Plot of the ratio $\frac{\mathbb{B}_t\left(\Theta_t^{(1)}\right)}{\mathbb{B}_t\left(\Theta_t^{(2)}\right)}$ for $d = 2, \Phi = 16$ for various values of $\sigma_f$ and $r$.

The full MASCOT algorithm as outlined in the application-specific context presented within this section is summarized in algorithm table 2.2.

## 2.7 Target Tracking with a Mobile Sensor Network

With recent advances in technology, the concept of a network of mobile autonomous units, each fitted with specialized sensing devices, cooperatively acting to track and follow some target, has become a physically achievable reality [30]. There are numerous benefits to employment of a mobilized suite of sensors to track a target, with perhaps the most important being the fact that the

*Complete the following steps for each time $t$:*

*S1:* For each agent $i$, collect information $\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right) = \left\{\psi\left(\mathbf{X}_{\mathbf{S}_{j,t-1}}\right) : j \neq i\right\}$

*S2:* For each agent $i$, make a prediction of the state $\mathbf{X}_t$ using $\mathbf{X}_{\mathbf{S}_{i,t-1}}$ and $\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)$

*S3:* For each possible candidate partition $\Theta_t^{(u)}$ do:

Determine the candidate sensor allocation for each agent, $\mathbf{Y}_{i,t}^{\Theta_t^{(u)}}$.

Determine the optimal sensor positions under this allocation, $\mathbf{q}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}}$ for each agent $i$.

Compute the objective,

$$\mathcal{O}_t\left(\Theta_t^{(u)}\right) = \sum_{i=1}^{G} \mathbb{D}_M\left(\mathbb{X}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) \mathbb{B}_i\left(\mathbf{X}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) + \lambda_1 \mathbb{D}\left(\Theta_t^{(u)}\right) \text{ where,}$$

$$\mathbb{B}_i\left(\mathbf{X}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) = \operatorname{tr}\left(\mathbb{E}\left[\mathbf{v}\mathbf{v}^\top\right]^{-1}\right), \quad \mathbf{v} = \frac{\partial \log f\left(\mathbf{Y}_{i,t} | \mathbf{X}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)}{\partial \mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}}$$

$$\mathbb{D}_M\left(\mathbb{X}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) = \exp\left(|\mathbf{S}_{i,t}|\right)$$

Choose $\Theta_t^{(\star)}$ yielding the minimum value for $\mathcal{O}_t$ as the partition for time $t$.

*S4:* Move all sensors to their calculated optimal positions under the allocation $\mathbf{q}_{\mathbf{S}_{i,t}}^{\Theta_t^{(\star)}}$.

*S5:* Collect measurements $\mathbf{Y}_{i,t}$ for each agent $i$.

*S6:* For each agent $i$,

For each target $k$ that satisfies $k \in \mathbf{S}_{i,t} \wedge k \in \mathbf{S}_{i,t-1}$ , draw $\mathbf{x}_{k,t}^{(m)} \sim \mathbf{A}\mathbf{x}_{k,t-1}^{(m)}$

For each target $k$ that satisfies $k \in \mathbf{S}_{i,t} \wedge k \notin \mathbf{S}_{i,t-1}$,

draw $\mathbf{x}_{k,t}^{(m)} \sim \mathcal{N}\left(\mathbf{A}\hat{\mathbf{x}}_{k,t-1}, \mathbf{A}\mathbf{C}_{k,t-1}\mathbf{A}^\top + \mathbf{Q}\right)$ where $[\hat{\mathbf{x}}_{k,t-1}, \mathbf{C}_{k,t-1}] \in \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)$

For each particle $\mathbf{X}_{\mathbf{S}_{i,t}}^{(m)} = \left\{\mathbf{x}_{k,t}^{(m)} : k \in \mathbf{S}_{i,t}\right\}, m = 1 \ldots M$ compute,

$w_{\mathbf{S}_{i,t}}^{(m)} = w_{\mathbf{S}_{i,t-1}}^{(m)} f\left(\mathbf{Y}_{i,t} \mid \mathbf{X}_{\mathbf{S}_{i,t}}^{(m)}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)$ where:

$$f\left(\mathbf{Y}_{i,t} \mid \mathbf{X}_{\mathbf{S}_{i,t}}^{(m)}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) \approx \prod_{j \in \mathbf{Y}_{i,t}} \mathcal{N}\left(\left\{\mathbf{Y}_{i,t}\right\}_j \mid h_j^{(s)}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{(m)}\right) + h_j^{(f)}\left(\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right), \left\{\sigma^{\star 2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j\right)$$

$$h_j^{(f)}\left(\hat{\mathbf{X}}_{\mathbf{F}_{i,t}}\right) = \sum_{k' \in \mathbf{F}_{i,t-1}} \Phi\left(\left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{k',t-1}\right)^{1:2} \right\|^{\alpha} + \epsilon\right)^{-1}$$

$$\left\{\sigma^{\star 2}\left(\mathbf{q}_{S_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j = \sigma_v^2 + \sum_{k' \in \mathbf{F}_{i,t-1}} \frac{\left\{\mathrm{D}_{\mathbf{S}_{i,t}}^{k'}\right\}_j \Phi^2 \alpha^2 \left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{k',t-1}\right)^{1:2} \right\|^{2\alpha-4}}{\left(\left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{k',t-1}\right)^{1:2} \right\|^{\alpha} + \epsilon\right)^4}$$

$$\left\{\mathrm{D}_{\mathbf{S}_{i,t}}^{k'}\right\}_j = \left(\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{k',t-1}\right)^{1:2}\right)^\top \left(\mathbf{A}\mathbf{C}_{k',t-1}\mathbf{A}^\top + \mathbf{Q}\right) \left(\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \left(\mathbf{A}\hat{\mathbf{x}}_{k',t-1}\right)^{1:2}\right)$$

Normalize the weights and if necessary, perform resampling *after S7.*

*S7:* For each agent i, form the information $\psi\left(\mathbf{X}_{\mathbf{S}_{i,t}}\right)$ that is then broadcast to other agents as:

$\psi\left(\mathbf{X}_{\mathbf{S}_{i,t}}\right) = \{[\hat{\mathbf{x}}_{k,t}, \mathbf{C}_{k,t}] : k \in \mathbf{S}_{i,t}\}$ where,

$$\hat{\mathbf{x}}_{k,t} = \sum_{m=1}^{M} w_{\mathbf{S}_{i,t}}^{(m)} \mathbf{x}_{k,t}^{(m)} \text{ and } \mathbf{C}_{k,t} = \sum_{m=1}^{M} w_{\mathbf{S}_{i,t}}^{(m)} \left(\hat{\mathbf{x}}_{k,t} - \mathbf{x}_{k,t}^{(m)}\right)\left(\hat{\mathbf{x}}_{k,t} - \mathbf{x}_{k,t}^{(m)}\right)^\top$$

**Table 2.2:** MASCOT Multi-Target Tracking PF Algorithm Implementation Summary

sensors can be dynamically positioned so as to extract the maximum possible information regarding the target state, thereby enhancing overall tracking performance. Over the years, this topic has become a very active area for research since actual implementation of such a system is not a trivial task. There are a large number of new engineering challenges faced, particularly involving sensor mobility and resource management. Determination of an optimal trajectory for the sensors can be an ambiguous task, which will in general depend on the specific optimality criteria chosen, the actual target track estimation methodology employed, and specific characteristics of the underlying target environment. A wide variety of solutions have been developed to handle sensor management, many with a sensible common theme of "information-driven" mobility as in [31],[32],[33], and [34], whereby sensor motion is carried out with the specific intent of optimizing the projected quality of measurements or the "information gain", at the new sensor locations. Other methods place an emphasis on a more balanced approach, paying specific attention to energy efficiency [35] or sensor coverage [36].

Current literature is relatively sparse in considering sensor positioning for an MTT environment. The work in [37] is a notable exception and provides a thorough investigation of the problem for a scenario involving sensors that provide range-only measurements corrupted by Gaussian noise. There, closed-form expressions are found for optimal sensor configurations with an arbitrary number of sensors or targets. While [37] can possibly be applied to RSSI sensor systems, it cannot be done so directly; one must first produce range estimates for each sensor-target pair from the RSSI measurements and act as if these estimates are themselves measurements that can be adequately modeled as having been corrupted by Gaussian noise. It is shown in the paper that the problem can be seen as a separable simultaneous optimization of criteria associated with each individual target, implying the interesting result that any optimal solution found for a group of targets will also be optimal for individual targets when considered in isolation.

There is a fundamental difference between that work and what will be presented. Here, sensor positioning is considered as a *joint task with estimation*, as can be seen in step 3 of 2.2. To be more precise, the optimality criteria for sensor positioning is based directly on the RSSI measurements. This is significantly more challenging due to the superpositionality of the sensors, however, it is believed to yield a significant benefit over the style of [37] since the affect of sensor positions on performance is more accurately represented. It will be clear that the results in [37] no longer hold true in this case; the optimal sensor configuration for a group of targets can be vastly different than optimums for isolated individual targets.

While the initial approach and model basis taken here for sensor trajectory-planning is based on [38], this paper diverges quickly due to the presence of interference, $\mathbf{X}_{\mathbf{F}_{i,t}}$ that will interfere with an agent's sensor measurements of $\mathbf{X}_{\mathbf{S}_{i,t}}$. It is the aim here to address the problem described; namely the investigation of optimal sensor positioning for a given agent's $\mathbf{X}_{\mathbf{S}_{i,t}}$, based directly on RSSI

measurements, and with the presence of an arbitrary number of interfering sources represented as $\mathbf{X}_{\mathbf{F}_{i,t}}$.

As mentioned, sensor positioning is done here as an integral part of MASCOT and thus it is sensible to choose an optimality criterion for this task that fits with the partitioning criterion embodied in equations (2.8) and (2.53). Rather than use the CRLB directly however, it has been chosen to maximize the determinant of the Fisher information matrix (FIM); this is termed D-optimality in current literature and is in contrast to A-optimality, which instead minimizes the trace of the CRLB matrix. A-optimality can be sensitive to scale changes in the problem, and although this is not an issue here, D-optimality does produce a simpler and more manageable form for the problem.

Let us then formulate the task of sensor position optimization within the MASCOT framework. Referring to Table 2.2, we wish to determine for each candidate partition, $\Theta_t^{(u)}$, the optimal sensor positions $\mathbf{q}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}}$ of allocated sensors $\mathbf{Y}_{i,t}^{\Theta_t^{(u)}}$ for the $i$-th agent. We thus have the optimization problem:

$$\bar{\mathbf{q}}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}} = \underset{\mathbf{q}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}}}{\operatorname{argmax}} \mathcal{G}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}} \qquad \mathcal{G}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}} \triangleq \det\left(\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}} \mathbf{v}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)},\top}\right]\right)$$
$$\text{subject to} \quad \| \left\{ \mathbf{q}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}} \right\}_k - \mathbf{x}_{l,t} \| \geq r \quad \text{for } k = 1 \ldots |\mathbf{q}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}}| \ \wedge \forall l \in \mathbf{S}_{i,t} \tag{2.74}$$

where the symbol[1] $\mathbf{v}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}}$ was defined in (2.56) and the symbol $|\mathbf{q}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}}|$ refers to the cardinality of this set. While not explicitly written in (2.74), we regard the objective function $\mathcal{G}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}}$ here as a function of $\mathbf{q}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}}$. Note the constraint that each sensor within $\mathbf{Y}_{i,t}^{\Theta_t^{(u)}}$ must maintain a minimum separation of $r$ from every target location within $\mathbf{X}_{\mathbf{S}_{i,t}}$.

With the stated assumptions on the communicated information $\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)$ that is described by (2.42), it was shown that $\left(\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}} \mathbf{v}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)},\top}\right]\right)$ takes the form in (2.65). To compute $\mathcal{G}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}}$ we must find the determinant of this $2|\mathbf{S}_{i,t}| \times 2|\mathbf{S}_{i,t}|$ matrix. While it is highly desirable to find a simplified form for this determinant, it represents a formidable task if $|\mathbf{S}_{i,t}| > 1$. What will be shown here is that a simple, intuitive form does exist for when there is a single target in $\mathbf{X}_{\mathbf{S}_{i,t}}$. Additionally, a method for generalizing to multiple targets in $\mathbf{X}_{\mathbf{S}_{i,t}}$ will be described and is solved for $|\mathbf{X}_{\mathbf{S}_{i,t}}| = 2$. The ultimate goal is to find a simplified form for an arbitrarily sized $\mathbf{X}_{\mathbf{S}_{i,t}}$, however this still remains a valuable area open to future investigation.

[1] The additional superscript $\Theta_t^{(u)}$ here simply means this term is for the given candidate partition $\Theta_t^{(u)}$

### 2.7.1 Positioning for Single-Target Estimation with Interference

Assume that a single target $\mathbf{x}_{l,t}$ is contained within $\mathbf{X}_{\mathbf{S}_{i,t}}$. The expression in (2.65) then simplifies to the sum of $|\mathbf{Y}_{i,t}|$ $2 \times 2$ matrices:

$$\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}} \mathbf{v}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)},\top}\right] = \alpha^2\Phi^2 \sum_{y_j \in \mathbf{Y}_{i,t}} \left( \frac{\left\| \left\{\mathbf{qs}_{i,t}\right\}_j - \mathbf{x}_{l,t}^{1:2} \right\|^{2(\alpha-2)}}{\left\{\sigma^{\star 2}\left(\mathbf{qs}_{i,t}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j \left( \left\| \left\{\mathbf{qs}_{i,t}\right\}_j - \mathbf{x}_{l,t}^{1:2} \right\|^{\alpha} + \epsilon \right)^4} \right.$$
$$\left. \times \begin{bmatrix} \left(\left\{\mathbf{qs}_{i,t}\right\}_{j^{(1)}} - \mathbf{x}_{l,t}^{(1)}\right)^2 & \left(\left\{\mathbf{qs}_{i,t}\right\}_{j^{(1)}} - \mathbf{x}_{l,t}^{(1)}\right)\left(\left\{\mathbf{qs}_{i,t}\right\}_{j^{(2)}} - \mathbf{x}_{l,t}^{(2)}\right) \\ \left(\left\{\mathbf{qs}_{i,t}\right\}_{j^{(1)}} - \mathbf{x}_{l,t}^{(1)}\right)\left(\left\{\mathbf{qs}_{i,t}\right\}_{j^{(2)}} - \mathbf{x}_{l,t}^{(2)}\right) & \left(\left\{\mathbf{qs}_{i,t}\right\}_{j^{(2)}} - \mathbf{x}_{l,t}^{(2)}\right)^2 \end{bmatrix} \right) \quad (2.75)$$

Then making the following definitions:

$$c_{j,t} = \frac{\Phi^2\alpha^2 \left\| \left\{\mathbf{qs}_{i,t}\right\}_j - \mathbf{x}_{l,t}^{1:2} \right\|^{2(\alpha-2)}}{\left\{\sigma^{\star 2}\left(\mathbf{qs}_{i,t}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j \left( \left\| \left\{\mathbf{qs}_{i,t}\right\}_j - \mathbf{x}_{l,t}^{1:2} \right\|^{\alpha} + \epsilon \right)^4} \quad (2.76)$$

$$\mathbf{d}_{j,t} = \begin{bmatrix} d_{j,1,t} \\ d_{j,2,t} \end{bmatrix} = \begin{bmatrix} \left(\left\{\mathbf{qs}_{i,t}\right\}_{j^{(1)}} - \mathbf{x}_{l,t}^{(1)}\right) \\ \left(\left\{\mathbf{qs}_{i,t}\right\}_{j^{(2)}} - \mathbf{x}_{l,t}^{(2)}\right) \end{bmatrix} \quad (2.77)$$

We can write:

$$\begin{aligned} \det\left(\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{\mathrm{i,t}}}^{\Theta_{\mathrm{t}}^{(\mathrm{u})}} \mathbf{v}_{\mathbf{S}_{\mathrm{i,t}}}^{\Theta_{\mathrm{t}}^{(\mathrm{u})},\top}\right]\right) &= \left(\sum_{y_j \in \mathbf{Y}_{i,t}} c_{j,t} d_{j,1,t}^2\right)\left(\sum_{y_j \in \mathbf{Y}_{i,t}} c_{j,t} d_{j,2,t}^2\right) - \left(\sum_{y_j \in \mathbf{Y}_{i,t}} c_{j,t} d_{j,1,t} d_{j,1,t}\right)^2 \\ &= \sum_{y_j, y_p \in \mathbf{Y}_{i,t}} c_{j,t} c_{p,t} \left(d_{j,1,t}^2 d_{p,2,t}^2 - d_{j,1,t} d_{j,2,t} d_{p,1,t} d_{p,2,t}\right) \\ &= \frac{1}{2} \sum_{y_j, y_p \in \mathbf{Y}_{i,t}} c_{j,t} c_{p,t} \|\mathbf{d}_{j,t} \otimes \mathbf{d}_{p,t}\|^2 \end{aligned} \quad (2.78)$$

where the symbol $\otimes$ is used to denote the outer product between two vectors.

Equation (2.78) is similar in form to equation (10) in [37]. Indeed, if we set $L = 0$ (no interferers), we end up with the following for a single target:

$$\det\left(\mathbb{E}\left[\mathbf{v}_{\mathbf{S}_{\mathrm{i,t}}}^{\Theta_{\mathrm{t}}^{(\mathrm{u})}} \mathbf{v}_{\mathbf{S}_{\mathrm{i,t}}}^{\Theta_{\mathrm{t}}^{(\mathrm{u})},\top}\right]\right) = \frac{\Phi^4\alpha^4}{2\sigma_{\mathrm{v}}^4} \sum_{\mathrm{y_j}, \mathrm{y_p} \in \mathbf{Y}_{\mathrm{i,t}}} \frac{\left(\|\mathbf{d}_{\mathrm{j,t}}\| \|\mathbf{d}_{\mathrm{p,t}}\|\right)^{2(\alpha-2)} \|\mathbf{d}_{\mathrm{j,t}} \otimes \mathbf{d}_{\mathrm{p,t}}\|^2}{\left(\left(\|\mathbf{d}_{\mathrm{j,t}}\|^{\alpha} + \epsilon\right)\left(\|\mathbf{d}_{\mathrm{p,t}}\|^{\alpha} + \epsilon\right)\right)^4} \quad (2.79)$$

It is fairly easy to prove (and has been done so rather elegantly in [37] even if under a somewhat different model) that the optimal sensor formation in this situation consists of the sensors uniformly distributed around a circle of radius $r$ centered at $\mathbf{x}_{l,t}$. It is also clear here that one can achieve any desired degree of precision by making $r$ arbitrarily small. It is not obvious if the same holds for (2.78); that if the inclusion of interference may place the optimal solution *outside*

*the active constraint region*, i.e., for the $j$-th optimal sensor location, $\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j$, this would mean $\left\|\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \mathbf{x}_{l,t}^{1:2}\right\| > r$. While no attempt to prove analytically that the radius constraint is active at an optimal solution will be made here, it is noted that individual terms in the sum tend to $\infty$ as $\left\|\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \mathbf{x}_{l,t}^{1:2}\right\| \to 0$. Regardless of the interference locations (as long as they are not co-located with the target), the objective function can be made arbitrarily large by tending each of these terms to zero. For this reason, for the remainder of this analysis, we will always assume the constraint is active for all sensors:

$$\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j = \mathbf{x}_{l,t}^{1:2} + r\left[\cos\left(\theta_{j,t}\right)\ \sin\left(\theta_{j,t}\right)\right]^\top \tag{2.80}$$

Complexity of the problem grows remarkably by the addition of interference [1] owing to several complicating factors. First and foremost, the $\left\{\sigma^{\star^2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j$ term in the denominator of $c_{j,t}$ renders the combined form of (2.78) intractable to a closed-form solution. Additionally, it can be seen that the interference creates "competing constraints" in the objective; minimization of the $\left\{\sigma^{\star^2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j$ terms can push the $\|\mathbf{d}_{j,t} \otimes \mathbf{d}_{p,t}\|^2$ terms away from their optimal values. It is rather obvious to see that this problem is generally nonconvex. Furthermore, it is noted that the objective function can be cast as a multiplicative programming problem, by maximizing $\exp\left[\mathcal{G}^{\Theta_t^{(u)}}\right]$, which is known to be $NP$-hard [39].

At this point, one could choose to proceed by selecting a numerical optimization technique to approach the problem. Although nonconvex, there does exist specific structure in the problem that can be exploited towards a viable solution. Namely, it can be recognized that while (2.78) is nonconvex, it can be decomposed into a sum-of-products of individually convex terms, i.e., the $j$-th term in the sum and the outer product are both convex over a restricted region. As a result, methods that make use of this property, namely [40] and [41], can be employed. Another option would be to apply an evolutionary algorithm, including techniques such as Genetic Algorithm [42], or Particle Swarm Optimization [43].

While the wide-range of numerical optimization options to choose from has been presented, it is of great desire to formulate a solution that is as computationally inexpensive as possible. The dynamic nature of the environment creates the need to adjust the sensor configuration at *every time step* for *every possible partition* $\Theta_t^{(u)}$. Also, having the optimal configuration under the chosen criteria will not in general guarantee optimal tracking performance; the CRLB is known to be a loose bound particularly for problems with severe nonlinearities [2] as is present in RSS measurements. As such, it is the goal here to formulate a fast, intuitive approach to sensor positioning that is relatively close to the optimum under the criteria chosen, without necessitating the high expense

[1] The addition of even one source eliminates the possibility of a closed-form solution.

[2] If the parameter $\epsilon$ is set to 0, the measurement model will have a singularity at $r = 0$

involved in the use of a numerical method. To this end, a solution is now presented that is based on a divide-and-conquer approach; each of the terms involved in (2.78) are considered separately, then their global interaction is considered.

It is clear that the $c_{j,t}$ terms in (2.78) represent a dominating factor in optimization of the objective. Indeed, if the outer product term were absent from this expression, the problem would consist solely of optimizing each sensor location with respect to the interference sources by minimizing $\left\{\sigma^{\star^2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j$ for all $j$. The outer product can in a way be seen to *force a compromise* to be made between optimality of each sensor w.r.t. the net interference configuration and *orthogonality* of the net sensor configuration. Let us for the moment neglect this compromise and focus on the expression for $\left\{\sigma^{\star^2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j$ given in (2.47) for the $j$-th sensor. We can first establish that in the presence of only a single interference source, the optimal location of sensor $j$ will be at the farthest point on the $r$-circle from the interference. To be more precise, with $\mathbf{F}_{i,t} = \{p\}$, (2.47) takes the form:

$$\left\{\sigma^{\star^2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j = \sigma_v^2 + \left(\frac{\Phi\alpha\sigma_p \left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \hat{\mathbf{x}}_{p,t}^{1:2}\right\|^{\alpha-1}}{\left(\left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \hat{\mathbf{x}}_{p,t}^{1:2}\right\|^{\alpha} + \epsilon\right)^2}\right)^2 \tag{2.81}$$

Notice that several simplifications were made in arriving at (2.81) from (2.47); namely it is assumed here that agent $i$ processes the information $\psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)$ described by (2.42) and forms the mean prediction $\hat{\mathbf{x}}_{p,t}$ and covariance estimate $\mathbf{C}_{p,t} \approx \sigma_p^2\mathbf{I}$ where,

$$\sigma_p^2 = \max\left(\left[\mathbf{A}\mathbf{C}_{p,t-1}\mathbf{A} + \mathbf{Q}\right]_{1,1}, \left[\mathbf{A}\mathbf{C}_{p,t-1}\mathbf{A}^\top + \mathbf{Q}\right]_{2,2}\right) \tag{2.82}$$

While this is mainly done here to simplify the subsequent analysis, it can be justified by arguing that this represents a conservative estimate of the true interference covariance prediction. Generalization of the following results to incorporate the true prediction remains an open area for future investigation.

It is fairly clear that (2.81) is minimized by maximizing $\left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \hat{\mathbf{x}}_{p,t}^{1:2}\right\|$. With the expression for $\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j$ as in (2.80) and expressing the interference location in polar coordinates as,

$$\hat{\mathbf{x}}_{p,t}^{1:2} = \mathbf{x}_{l,t}^{1:2} + \left[\hat{r}_{p,t}\cos\hat{\lambda}_{p,t} \quad \hat{r}_{p,t}\sin\hat{\lambda}_{p,t}\right]^\top \tag{2.83}$$

we have:

$$\left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \hat{\mathbf{x}}_{p,t}^{1:2}\right\| = \sqrt{r^2 + \hat{r}_{p,t}^2 - 2r\hat{r}_{p,t}cos\left(\theta_{j,t} - \hat{\lambda}_{p,t}\right)} \tag{2.84}$$

which is maximized by choosing $\theta_{j,t} = \hat{r}_{p,t} \pm \pi$. The situation is considerably more complicated for more than one interference source; there is now an interaction between all the sensor-interference distances, i.e., the farthest point from one interferer may be detrimentally close to another. One

possible solution that is proposed here is to optimize a weighted combination of the sensor-interferer distances, namely one can formulate a solution by solving the problem:

$$\begin{aligned}\tilde{\theta}_{j,t} = \underset{\theta_{j,t}}{\operatorname{argmax}} \sum_{p\in\mathbf{F}_{i,t}} g_{p,t} \left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \hat{\mathbf{x}}_{p,t}^{1:2} \right\|^2 \\ \text{subject to} \quad \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j = \mathbf{x}_{l,t}^{1:2} + \begin{bmatrix} r\cos\theta_{j,t} & r\sin\theta_{j,t}\end{bmatrix}^\top \end{aligned} \tag{2.85}$$

where $g_{p,t}$ are weighting coefficients chosen to affect the solution's optimality with respect to minimizing (2.47). With $\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j$ expressed as (2.84), it can be shown that the solution to (2.85) is:

$$\tilde{\theta}_{j,t} = \operatorname{atan2}\left( \sum_{p\in\mathbf{F}_{i,t}} g_{p,t}\hat{r}_{p,t}\sin\left(\hat{\lambda}_{p,t}\right), \sum_{p\in\mathbf{F}_{i,t}} g_{p,t}\hat{r}_{p,t}\cos\left(\hat{\lambda}_{p,t}\right)\right) \tag{2.86}$$

We can then select a solution to minimize (2.47) as,

$$\theta_{j,t} = \frac{n\pi}{2} + \tilde{\theta}_{j,t}\,, \quad n = \underset{n\in\mathcal{N}}{\operatorname{argmin}}\, \sigma^\star\left\{\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t}}\right)\right)\right\}_j \left( r\begin{bmatrix}\cos\left(\frac{n\pi}{2} + \tilde{\theta}_{j,t}\right) \\ \cos\left(\frac{n\pi}{2} + \tilde{\theta}_{j,t}\right)\end{bmatrix}\right) \tag{2.87}$$

The reason this configuration can work reasonably well is intuitive; the objective in (2.85) ensures the sensor is not positioned too close on the $r$-circle to any given interferer. The weighting coefficients can be chosen in a number of different ways; one possibility can be to choose them such that each term in (2.85) forms a "linear" approximation to each term within (2.47). Another would be to simply assign the maximum weight to the term involving the interference source closest to the target location. This solution is herein labeled as BASIC and has been found to yield fair performance considering its simplicity.

The second method proposed here is to form a piecewise-linear approximation to (2.47). In what follows, it is assumed the parameter values $\alpha = 2$ and $\epsilon = 0$ for the measurement model in (2.20). While the results can be extended to cover more general values, the derivation is considerably lengthier and has not been fully attempted yet. The form for (2.47) then becomes:

$$\left\{\sigma^{\star^2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j = \sigma_v^2 + \sum_{p\in\mathbf{F}_{i,t}} \left( \frac{\Phi^2\alpha^2\sigma_p^2}{\left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \hat{\mathbf{x}}_{p,t}^{1:2}\right\|^6} \right) \tag{2.88}$$

It is proposed to approximate each term of the sum in (2.88) as follows:

$$\sigma_{j,p,t}^{\star^2}\left(\theta_{j,t}\right) \triangleq \left( \frac{\Phi^2\alpha^2\sigma_p^2}{\left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \hat{\mathbf{x}}_{p,t}^{1:2}\right\|^6} \right) \approx m_{j,p,t}^\star\left(\theta_{j,t}\right)\left(\theta_{j,t} - \hat{\lambda}_{p,t}\right) + b_{j,p,t}^\star\left(\theta_{j,t}\right) \tag{2.89}$$

With the functions $m^{\star}_{j,p,t}$ and $b^{\star}_{j,p,t}$ defined as:

$$\begin{aligned} b^{\star}_{j,p,t}\left(\theta_{j,t}\right) &= \sum_{p\in\mathbf{F}_{i,t}} b_{j,p,t}\mathbb{1}\left[\theta^{-}_{j,p,t} \leq \theta_{j,t} < \theta^{+}_{j,p,t}\right] \\ m^{\star}_{j,p,t}\left(\theta_{j,t}\right) &= \sum_{p\in\mathbf{F}_{i,t}} m_{j,p,t}\left(\mathbb{1}\left[\theta^{-}_{j,p,t} \leq \theta_{j,t} < \hat{\lambda}_{p,t}\right] - \mathbb{1}\left[\hat{\lambda}_{p,t} \leq \theta_{j,t} < \theta^{+}_{j,p,t}\right]\right) \end{aligned} \tag{2.90}$$

where $\mathbb{1}\left[x\right]$ denotes the indicator function for the argument and where the individual slope constant $m_{j,p,t}$ is the scaled magnitude[1] of the derivative evaluated at the point $\theta^{H}_{j,p,t}$ where $\sigma^{\star^2}_{j,p,t}\left(\theta_{j,t}\right)$ takes on half its maximum value:

$$\begin{aligned} \theta^{H}_{j,p,t} &= \hat{\lambda}_{p,t} + \arccos\left(\frac{\left(1-2^{\frac{1}{3}}\right)\left(r^2+\hat{\lambda}^2_{p,t}\right)+2^{\frac{2}{3}}r\hat{\lambda}_{p,t}}{2r\hat{\lambda}_{p,t}}\right) \\ m_{j,p,t} &= c\left|-\frac{6\phi^2\alpha^2\sigma^2_l r\hat{r}_{p,t}\sin\left(\theta^{H}_{j,p,t}-\hat{\lambda}_{p,t}\right)}{\left(r^2+\hat{r}^2_{p,t}-2r\hat{r}_{p,t}\cos\left(\theta^{H}_{j,p,t}-\hat{\lambda}_{p,t}\right)\right)^4}\right| \end{aligned} \tag{2.91}$$

and the remaining terms are defined as:

$$\begin{aligned} b_{j,p,t} &= \frac{\Phi^2\alpha^2\sigma^2_l}{\left(r^2+\hat{\lambda}^2_{p,t}-2r\hat{\lambda}_{p,t}\right)^3} \\ \theta^{\pm}_{j,p,t} &= \hat{\lambda}_{p,t} \pm \frac{b_{j,p,t}}{m_{j,p,t}} \end{aligned} \tag{2.92}$$

Note that all points of $\theta_{j,t}$ here are understood to be with respect to modulo $2\pi$.

An approximation to the minimum of (2.47) is then easily found by minimizing over each distinct segment of the piecewise function. Namely, the set of all points $\left(\theta^{-}_{j,p,t}, \hat{\lambda}_{p,t}, \theta^{+}_{j,p,t}\right)$ are arranged into a sorted array $\Psi_t$. The $z$-th non-overlapping line segment formed by two adjacent points of this array is labeled $\Psi_{z,t}$, written as $\left[\Psi^{(-)}_{z,t}, \Psi^{(+)}_{z,t}\right]$. The critical point of each segment, denoted as $\Psi^{(\star)}_{z,t}$ is then declared to be:

$$\Psi^{(\star)}_{z,t} = \begin{cases} \Psi^{(-)}_{z,t} & \text{if } \sum\limits_{p\in\mathbf{S}_{i,t}} m^{\star}_{j,p,t}\left(\Psi^{(-)}_{z,t}\right) > 0 \\ \Psi^{(+)}_{z,t} & \text{if } \sum\limits_{p\in\mathbf{S}_{i,t}} m^{\star}_{j,p,t}\left(\Psi^{(-)}_{z,t}\right) < 0 \\ \frac{1}{2}\left(\Psi^{(+)}_{z,t}+\Psi^{(-)}_{z,t}\right) & \text{if } \sum\limits_{p\in\mathbf{S}_{i,t}} m^{\star}_{j,p,t}\left(\Psi^{(-)}_{z,t}\right) = 0 \end{cases} \tag{2.93}$$

The optimal point is then simply $\text{argmin}_z\, \sigma^{(\star^2)}\left(\Psi^{(\star)}_{z,t}\right)$

An example scenario was simulated with two separate runs, the first consisting of two interferers ($L=2$) and the second with six ($L=6$). In both cases, $\hat{r}_{p,t}=2$ for all $p$ except the first, for

[1] With $c=1$ the approximation tends to underestimate. Substantial improvement has been found with $c=\frac{1}{\sqrt{2}}$. The half-max point was chosen as it models the "region of significant influence" of each interferer well.

which $\hat{r}_{2,t} = 1.5$. The values for $\hat{\lambda}_{p,t}$ were fixed at $0, \frac{\pi}{2}, \frac{\pi}{8}$,and $\frac{3\pi}{2}$, and $\pi$ for the second through sixth interferers. The value for $\hat{\lambda}_{1,t}$ was varied uniformly over 0 to $2\pi$ for 50 different values, with each marking a separate trial of the given run (and producing different objective values). The value for $\hat{\lambda}_1$ was varied uniformly over 0 to $2\pi$ for 50 different values, with each marking a separate trial of the given run (and producing different objective values). The remaining parameter values were fixed at $M = K$, $\alpha = 2$, $\epsilon = 0$, $\Phi = 10$, $\sigma_v^2 = 0.01$, and $\sigma_l^2 = 0.01$ for all $l$. Figure 2.5 shows the objective value for the two proposed solutions, Basic and Piecewise, along with a numerically computed optimal which was found by conducting a line-search over 1000 points within the closed interval $[0, 2\pi]$ at each distinct configuration. It is interesting to note that Basic performs reasonably well

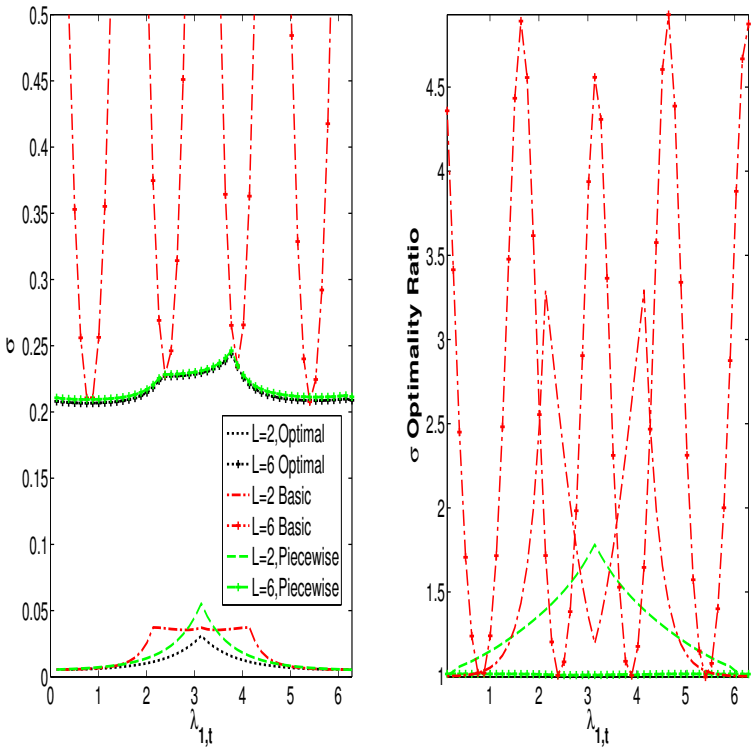


**Figure 2.5:** Comparison of proposed solutions for minimizig the term $\sigma_k \left(\mathbf{s}_{k,}\right)^2$ with respect to the $k$-th sensor angle.

for $L = 2$ but rapidly degrades at $L = 6$, whereas Piecewise maintains adequate performance in both scenarios.

Let us now consider how the aforementioned results can be used to form a viable solution towards optimization of the original objective function in (2.78) for an arbitrary number of sensors. While the Piecewise algorithm's final result is to provide an approximation to the global minimum value for $\left\{\sigma^\star\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j$ it also partitions this function into $3L$ distinct line segments, each with their own local minimum. We can use this information to our advantage; namely we can maximize the original partition over a discrete set of $|\mathbf{Y}_{i,t}|$-tuples derived from the set $\Psi_t$ as opposed

| |
|---|
| Step 1: For each segment $\Psi_i \in \Psi$ form $\left\{\Psi_i^{(m)}\right\}_{m=1:M}$ |
| where, $\Psi_i^{(m)} = \Psi_i^{(-)} + \max\left(\frac{m\pi}{K}, \frac{m\left(\Psi_i^{(+)} - \Psi_i^{(-)}\right)}{M}\right)$. |
| Step 2: Augment $\Psi$ to $\Psi^{(a)}$ with all $\left\{\Psi_i^{(m)}\right\}_{m=1:M}$ |
| Step 3: Choose $\theta_1$ as $\text{argmin}_{\Psi^{(a)}}\ \sigma^{(\star)}\left(\Psi^{(a)}\right)$. Set $k = 2$. |
| Step 4: Form all $K$-tuples $\hat{\theta}_\tau = \{\theta_{1:k-1}, \tau\}$ with $\tau \in \Psi^{(a)}$. |
| Step 5: Compute $\mathcal{G}^{(\tau)}$ for the $k$ sensor arrangement $\hat{\theta}_\tau$. |
| Step 6: Set $\theta_{1:k} = \text{argmax}_{\hat{\theta}_\tau}\ \mathcal{G}^{(\tau)}$. If $k < K$ go to Step 3. |

**Table 2.3:** Fast-Piecewise Algorithm Summary

to maximization over $\mathbb{R}^{|\mathbf{Y}_{i,t}|}$. The full algorithm which accomplishes this is labeled Fast-Piecewise and is outlined in table 2.3.

Performance of Fast-Piecewise can be compared with two well-known numerical methods, namely the Quasi-Newton method along with the Genetic Algorithm. The same example scenario (and same parameter set) described for evaluating the single-sensor Piecewise algorithm was used here, however this time the number of sensors were also varied. The log of the objective function $\mathcal{G}_{\mathbf{S}_{i,t}}$ (labeled as O in the figure) at each different position of $\hat{\lambda}_{1,t}$ was plotted, again keeping the remaining interferers fixed at specific positions; the results are illustrated in figure 2.6. It is seen that Fast-Piecewise performs equally well to the alternative techniques yet is also dramatically more efficient, with computation speeds as high as 60X that of GA observed.

Notice that little has been said yet regarding the first method initially mentioned, referred to as Basic. It is not as clear how one can apply this technique of minimizing $\left\{\sigma^{\star^2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j$ for a single sensor, towards optimization of the full objective function $\mathcal{G}_{\mathbf{S}_{i,t}}^{\Theta_t^{(u)}}$; the weighting coefficients in (2.85) are somewhat arbitrary and it is unclear how to select them in a general multi-sensor case. However, if the target environment is such that only a single interference term dominates the expression in (2.47), we can make the approximation:

$$\left\{\sigma^{\star^2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j \approx \sigma_v^2 + \frac{\left\{\mathrm{D}_{\mathbf{S}_{i,t}}^{\eta}\right\}_j \Phi^2\alpha^2 \left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \hat{\mathbf{x}}_{\eta,t}^{1:2} \right\|^{2\alpha-4}}{\left( \left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \hat{\mathbf{x}}_{\eta,t}^{1:2} \right\|^{\alpha} + \epsilon \right)^4} \tag{2.94}$$

where it is again assumed that all sensor locations $\mathbf{q}_{\mathbf{S}_{i,t}}$ are positioned about a circle of radius $r$ centered at the single TOI, $\mathbf{x}_{l,t} \in \mathbf{S}_{i,t}$, and the index $\eta$ is chosen as [1] ,

$$\eta = \underset{\eta \in \mathbf{F}_{i,t}}{\text{argmin}} \left\| \mathbf{x}_{l,t}^{1:2} - \hat{\mathbf{x}}_{\eta,t}^{1:2} \right\| \tag{2.95}$$

[1] Another possibilty is to choose the interferer that would contribute the maximum amount of signal to the measurement if the sensor were positioned as close as possible to it on the $r$-circle

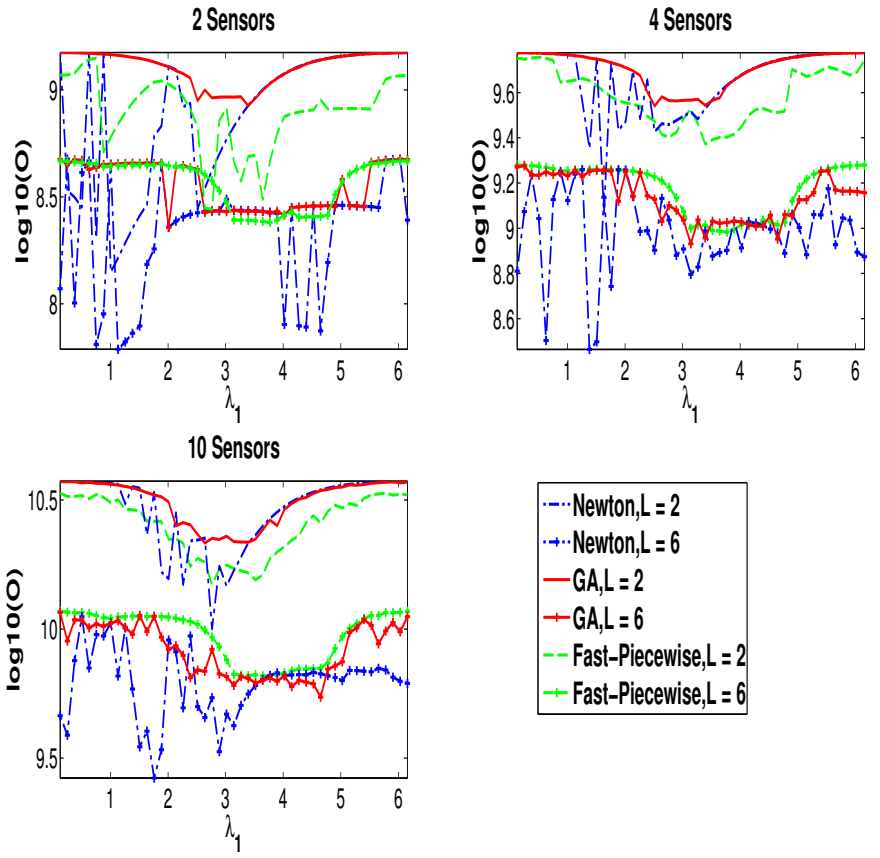


**Figure 2.6:** Comparison of proposed solution to numerical/evolutionary techniques.

Essentially, all interference terms are neglected except for the one that is closest to the TOI. With this simplification, we can make use of the result from (2.84) to form a solution for multiple sensors. To see how this can be done, let us momentarily neglect the interference contribution; than our net objective will be to maximize the sum of cross-products between all sensors in $\mathbf{Y}_{i,t}$. As discussed earlier, it can easily be shown that this sum will be maximimum for sensors configurations on the $r$-circle that yield an *orthogonal system*; expressing the term $\|\mathbf{d}_{j,t} \otimes \mathbf{d}_{p,t}\|^2$ in (2.78) as $r^2 \sin(\theta_{j,t} - \theta_{p,t})^2$, assuming $|\mathbf{Y}_{i,t}| = K$ we would like to find critical points to maximize the sum[1] by solving the set of equations,

$$\frac{\partial}{\partial \theta_{k,t}} \left( \sum_{k,p}^{K} \sin(\theta_{k,t} - \theta_{p,t})^2 \right) = 0 \qquad k = 1 \ldots K \tag{2.96}$$

This yields the condition for all k,

$$\sum_{k=1}^{K} \sin(\theta_{k,t} - \theta_{p,t}) \cos(\theta_{k,t} - \theta_{p,t}) = 0 \tag{2.97}$$

[1]The $c_{j,t}$ terms are constant here since we are assuming the interference can be neglected

Leveraging the well-known identity,

$$\sum_{i=0}^{K-1} \cos\left(\frac{2\pi}{K}i\right) \sin\left(\frac{2\pi}{K}i\right) = 0 \tag{2.98}$$

yields the result stated in [37], that one possible optimal configuration consists of distributing sensors evenly around the $r$-circle, i.e., $\theta_{k,t} = \xi + \frac{2\pi}{K}i$ for $i = 0 \dots K-1$ and $\xi \in [0\, 2\pi]$. It can also be shown that the following identity also holds,

$$\sum_{i=0}^{K-1} \cos\left(\frac{\pi}{K}i\right) \sin\left(\frac{\pi}{K}i\right) = 0 \tag{2.99}$$

This suggests an alternative optimal configuration, whereby sensors are evenly distributed around a half-circle, which will fit well with joint optimization for the interference component. To be precise, we can use this configuration to satisfy "unweighted" optimality[1] for the cross-product terms while positioning sensors as far away as possible from the interferer. This can be done by selecting the sensor angles as,

$$\theta_{k,t} = \angle(\mathbf{x}_{l,t}, \mathbf{x}_{\eta,t}) + \pi\left(1 - \frac{K-1}{2K}\right) + \frac{\pi}{K}i\,, \qquad i = 1 \dots K \tag{2.100}$$

where $\angle(\mathbf{x}_{l,t}, \hat{\mathbf{x}}_{\eta,t})$ denotes the angle between the TOI, $\mathbf{x}_{l,t}$ and $\mathbf{x}_{\eta,t}$ the interference location. This configuration spaces the sensors with maximum orthogonality but rotates the half circle to *face away* from the interference, providing the maximum separation. This solution is herein labeled Max-Sep and it is obvious this represents the simplest and maximally efficient solution.

A scenario was setup in simulation to compare performance between GA, Fast-Piecewise, and Max-Sep in specifically varying sensor configurations where $K = 4$. Namely, four interference sources are located centered about the target, each with radius $D$ except the first whose radius $= 1.5$. The first interferer angle, $\angle(\mathbf{x}_{l,t}, \hat{\mathbf{x}}_{1,t})$ is uniformly varied over the interval $[02\pi]$, while the remaining interferer angles are $\begin{bmatrix}0 & \pi/2 & \pi\end{bmatrix}$. Remaining parameters include the tracking radius, $r = 1$ for all sensors, $\Phi = 10$, and $\sigma_p = 0.1$ for all interferers. The log-ratio of the Fast-Piecewise and Max-Sep solution objectives to GA are plotted in figure 2.7; and as can be expected, the performance of Max-Sep deteriorates as the "dominating source" assumption breaks down (as $D$ decreases), but does perform fairly when it holds. It is finally interesting to note that Max-Sep can in fact perform similarly or *better* than Fast-Piecewise when the assumption holds strongly. Figure 2.8 illustrates the same scenario, however this time $r = 6$, the first interferer rotates about $D = 8.5$, and the remaining interferers are positioned at much farther distances. We can immediately see from the axis scale that both algorithms exhibit nearly identical performance as GA; there is also little discernible difference between the two suboptimal solutions. This suggests a hybrid approach whereby Fast-Piecewise is employed when multiple interferers are relatively close to the TOI, and Max-Sep otherwise.

[1] The sum would be optimal if the interference "weighting" terms $c_{k,t}$ were not present

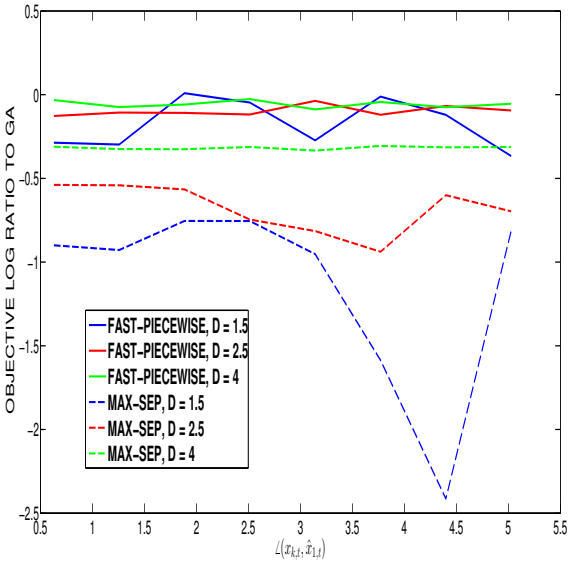


**Figure 2.7:** Comparison of two suboptimal solutions to GA over varying target-interference separation.

### 2.7.2 Positioning for Joint Estimation of Two Targets

The case of optimal sensor positioning for the most general scenario; whereby there are multiple interferers affecting sensor measurements of $\mathbf{X}_{\mathbf{S}_{i,t}}$ that consists of more than one target, is vastly more complicated than the preceding development. This complication is due to the need of finding the determinant of the $2|\mathbf{S}_{i,t}| \times 2|\mathbf{S}_{i,t}|$ matrix involved in the objective described in (2.74). For the case of $|\mathbf{S}_{i,t}| = 2$ with $\alpha = 2$ and $\epsilon = 0$[1] if one attempt's direct simplification of the determinant, a compact informative form can be attained. However, an alternative formulation of the problem at hand based on the results in [44] will be posed that elucidates how one may obtain a closed form expression for arbitrarily sized $\mathbf{S}_{i,t}$[2].

Referring back to the approximation for $f(\mathbf{Y}_{i,t}|\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right))$ in (2.45), this can be rewritten as,[3]

$$f(\mathbf{Y}_{i,t}|\mathbf{X}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)) \approx (2\pi)^{-|\mathbf{Y}_{i,t}|/2} \det\left[\mathbf{R}_{\mathbf{S}_{\mathrm{i,t}}}\right]^{-1/2} \exp\left\{-\frac{1}{2}\left[\bar{\mathbf{y}}_{\mathrm{i,t}} - \mathbf{h}\left(\mathbf{X}_{\mathrm{t}}\right)\right]^{\top} \mathbf{R}^{-1}_{\mathbf{S}_{\mathrm{i,t}}}\left[\bar{\mathbf{y}}_{\mathrm{i,t}} - \mathbf{h}\left(\mathbf{X}_{\mathrm{t}}\right)\right]\right] \quad (2.101)$$

where $\bar{\mathbf{y}}_{i,t}$ here refers to the $|\mathbf{Y}_{i,t}| \times 1$ column vector of vertically-stacked elements in $\mathbf{Y}_{i,t}$. $\mathbf{R}_{\mathbf{S}_{i,t}}$ is the $|\mathbf{Y}_{i,t}| \times |\mathbf{Y}_{i,t}|$ covariance matrix of the measurement set for agent $i$, and $\mathbf{h}\left(\mathbf{X}_t\right)$ is the $|\mathbf{Y}_{i,t}| \times 1$

[1] General values for $\alpha$ and $\epsilon$ were not attempted but it is expected that one can arrive at a similar form using the method described

[2] This is of great value for not only sensor positioning, but also for efficient computation of the partitioning objective function as described in in 2.2

[3] Note the superscript is intentionally dropped in $\mathbf{X}^{(m)}_{\mathbf{S}_{i,t}}$ for clarity (it is assumed $\mathbf{X}_{\mathbf{S}_{i,t}}$ is "known" here)

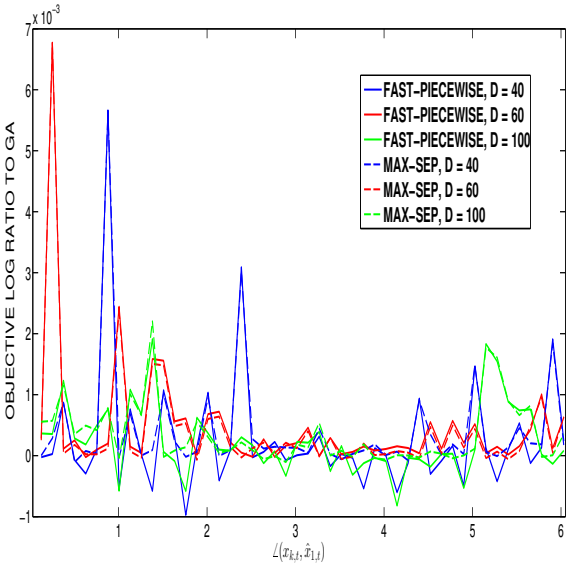


**Figure 2.8:** Comparison to GA over larger target-interference separations.

vector as follows:

$$\mathbf{R}_{\mathbf{S}_{i,t}} = \mathbb{E}\left[\bar{\mathbf{y}}_{i,t}\bar{\mathbf{y}}_{i,t}^{\top}\right] = \mathrm{diag}\left(\sigma_1^{\star^2}, \sigma_2^{\star^2}, \ldots, \sigma_{|\mathbf{Y}_{i,t}|}^{\star^2}\right)$$
$$[\mathbf{h}(\mathbf{X}_t)]_k = \sum_{j=1}^{L} \frac{\Phi}{\left\| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_k - \mathbf{x}_{j,t}^{1:2} \right\|^{\alpha} + \epsilon} \qquad k = 1 \ldots |\mathbf{Y}_{i,t}| \tag{2.102}$$

Notice the shorthand $\sigma_j^{\star^2} = \left\{\sigma^{\star^2}\left(\mathbf{q}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right\}_j$ is used here. The FIM that was defined in (2.55) can then be rewritten as,

$$\mathbb{J}^{-1}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2} \mid \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) = \mathbb{E}\left[\left(\frac{\partial}{\partial \mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}} \log f\left(\mathbf{Y}_{i,t} \mid \mathbf{X}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right)\left(\frac{\partial}{\partial \mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}} \log f\left(\mathbf{Y}_{i,t} \mid \mathbf{X}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)\right)^{\top}\right] \tag{2.103}$$

Where for a column vector $\mathbf{v}$ and scalar function $f(\mathbf{v})$, the notation $\frac{\partial f(\mathbf{v})}{\partial \mathbf{v}}$ is understood to refer to the column vector with elements $\left[\frac{\partial f(\mathbf{v})}{\partial \mathbf{v}}\right]_k = \frac{\partial}{\partial [\mathbf{v}]_k} f(\mathbf{v})$. The likelihood derivative vector in equation (2.103) is,

$$\frac{\partial}{\partial \mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}} \log f\left(\mathbf{Y}_{i,t} \mid \mathbf{X}_{\mathbf{S}_{i,t}}, \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) = \left[\frac{\partial}{\partial \mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}} \mathbf{h}^{\top}(\mathbf{X}_t)\right] \mathbf{R}_{\mathbf{S}_{i,t}}^{-1}\left[\bar{\mathbf{y}}_{i,t} - \mathbf{h}(\mathbf{X}_t)\right] \tag{2.104}$$

Allowing the FIM to be written as,

$$\mathbf{J}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2} | \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) = \mathbf{G}_{\mathbf{S}_{i,t}}^{\top} \mathbf{R}_{\mathbf{S}_{i,t}}^{-1} \mathbf{G}_{\mathbf{S}_{i,t}} \tag{2.105}$$

where the matrix $\mathbf{G}_{\mathbf{S}_{i,t},t}$ is,

$$\mathbf{G}_{\mathbf{S}_{i,t},t} = \left[\mathbf{g}_t^{(1)}, \mathbf{g}_t^{(2)}, \ldots, \mathbf{g}_t^{2|\mathbf{S}_{i,t}|}\right] \qquad \text{with} \qquad \mathbf{g}_t^{(k)} = \frac{\partial}{\partial \left[\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}\right]_k} \mathbf{h}\left(\mathbf{X}_t\right) \tag{2.106}$$

Let us then partition the vector $\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2}$ according to individual targets allowing us to form a partition of $\mathbf{J}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2} | \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)$ as,

$$\begin{bmatrix} \mathbf{G}_{1,t}^\top \mathbf{R}_{\mathbf{S}_{i,t}}^{-1} \mathbf{G}_{1,t} & \mathbf{G}_{1,t}^\top \mathbf{R}_{\mathbf{S}_{i,t}}^{-1} \mathbf{G}_{2,t} & \ldots & \mathbf{G}_{1,t}^\top \mathbf{R}_{\mathbf{S}_{i,t}}^{-1} \mathbf{G}_{|\mathbf{S}_{i,t}|,t} \\ \mathbf{G}_{2,t}^\top \mathbf{R}_{\mathbf{S}_{i,t}}^{-1} \mathbf{G}_{1,t} & \mathbf{G}_{2,t}^\top \mathbf{R}_{\mathbf{S}_{i,t}}^{-1} \mathbf{G}_{2,t} & \ldots & \\ \vdots & & \ddots & \\ \mathbf{G}_{|\mathbf{S}_{i,t}|,t}^\top \mathbf{R}_{\mathbf{S}_{i,t}}^{-1} \mathbf{G}_{1,t} & \ldots & & \mathbf{G}_{|\mathbf{S}_{i,t}|,t}^\top \mathbf{R}_{\mathbf{S}_{i,t}}^{-1} \mathbf{G}_{|\mathbf{S}_{i,t}|,t} \end{bmatrix} \tag{2.107}$$

where here the matrix $\mathbf{G}_{k,t}$ is defined as,

$$\mathbf{G}_{k,t} = \left[\frac{\partial}{\partial \left[\mathbf{x}_{k,t}\right]_1} \mathbf{h}\left(\mathbf{X}_t\right), \frac{\partial}{\partial \left[\mathbf{x}_{k,t}\right]_2} \mathbf{h}\left(\mathbf{X}_t\right)\right] = \begin{bmatrix} \mathbf{u}_{1,k,t}^{(1)} & \mathbf{u}_{1,k,t}^{(2)} \\ \mathbf{u}_{2,k,t}^{(1)} & \mathbf{u}_{2,k,t}^{(2)} \\ \vdots & \vdots \\ \mathbf{u}_{|\mathbf{Y}_{i,t}|,k,t}^{(1)} & \mathbf{u}_{|\mathbf{Y}_{i,t}|,k,t}^{(2)} \end{bmatrix} \quad \text{for } k \in \mathbf{S}_{i,t} \tag{2.108}$$

and where the vector $\mathbf{u}_{j,k,t} = \frac{1}{r_{j,k,t}^4}\left[\mathbf{d}_{j,k,t}^{(1)}, \mathbf{d}_{j,k,t}^{(2)}\right]^\top$ is defined using,

$$\mathbf{d}_{j,k,t} \triangleq \begin{bmatrix} \left(\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_{j^{(1)}} - \mathbf{x}_{k,t}^{(1)}\right) \\ \left(\left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_{j^{(2)}} - \mathbf{x}_{k,t}^{(2)}\right) \end{bmatrix} \qquad r_{j,k,t} \triangleq \| \left\{\mathbf{q}_{\mathbf{S}_{i,t}}\right\}_j - \mathbf{x}_{k,t} \| \tag{2.109}$$

In what follows, the $t$ subscript is dropped; it should be clear from context which variables depend on time. To simplify the subsequent derivation, the matrix $\bar{\mathbf{G}}_k$ is defined as,

$$\bar{\mathbf{G}}_k = \mathbf{R}_{\mathbf{S}_i}^{-1/2} \mathbf{G}_k \qquad \text{where} \qquad \mathbf{R}_{\mathbf{S}_i}^{-1/2} = \text{diag}\left(\frac{1}{\sigma_1^\star}, \frac{1}{\sigma_2^\star}, \ldots, \frac{1}{\sigma_{|\mathbf{Y}_i|}^\star}\right) \tag{2.110}$$

In the case where $|\mathbf{S}_{i,t}| = 2$, we can form the LDL decomposition of $\mathbf{J}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2} | \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right)$ as,

$$\mathbf{J}\left(\mathbf{X}_{\mathbf{S}_{i,t}}^{1:2} | \psi\left(\mathbf{X}_{\mathbf{F}_{i,t-1}}\right)\right) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \bar{\mathbf{G}}_2^\top \left(\bar{\mathbf{G}}^*\right)^\top & \mathbf{I} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{G}}_1^\top \bar{\mathbf{G}}_1 & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{G}}_2^\top \left(\mathbf{I} - \mathbf{P}_{G_1}\right) \bar{\mathbf{G}}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{G}_1^* \bar{\mathbf{G}}_2 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{2.111}$$

where $\mathbf{G}_1^* = \left(\bar{\mathbf{G}}_1^\top \bar{\mathbf{G}}_1\right)^{-1} \bar{\mathbf{G}}_1^\top$ is the pseudoinverse of $\bar{\mathbf{G}}_1$ and $\mathbf{P}_{G_1} = \bar{\mathbf{G}}_1 \left(\bar{\mathbf{G}}_1^\top \mathbf{G}_1\right)^{-1} \bar{\mathbf{G}}_1^\top$ is the orthogonal projection onto the linear subspace generated by $\bar{\mathbf{G}}_1$. An expression for the matrix $\mathbf{P}_{G_1}$ can be obtained by first noting that,

$$\bar{\mathbf{G}}_1^\top \bar{\mathbf{G}}_1 = \begin{bmatrix} \sum_{j=1}^{|\mathbf{Y}_{i,t}|} \left(\frac{\mathbf{u}_{j,1}^{(1)}}{\sigma_j^\star}\right)^2 & \sum_{j=1}^{|\mathbf{Y}_{i,t}|} \frac{\mathbf{u}_{j,1}^{(1)} \mathbf{u}_{j,1}^{(2)}}{\sigma_j^{\star 2}} \\ \sum_{j=1}^{|\mathbf{Y}_{i,t}|} \frac{\mathbf{u}_{j,1}^{(1)} \mathbf{u}_{j,1}^{(2)}}{\sigma_j^{\star 2}} & \sum_{j=1}^{|\mathbf{Y}_{i,t}|} \left(\frac{\mathbf{u}_{j,1}^{(2)}}{\sigma_j^\star}\right)^2 \end{bmatrix} \tag{2.112}$$

$$\left(\bar{\mathbf{G}}_1^\top \bar{\mathbf{G}}_1\right)^{-1} = \frac{1}{D_1}\begin{bmatrix} \sum_{j=1}^{|\mathbf{Y}_{i,t}|}\left(\frac{\mathbf{u}_{j,1}^{(2)}}{\sigma_j^\star}\right)^2 & -\sum_{j=1}^{|\mathbf{Y}_{i,t}|}\frac{\mathbf{u}_{j,1}^{(1)}\mathbf{u}_{j,1}^{(2)}}{\sigma_j^{\star^2}} \\ -\sum_{j=1}^{|\mathbf{Y}_{i,t}|}\frac{\mathbf{u}_{j,1}^{(1)}\mathbf{u}_{j,1}^{(2)}}{\sigma_j^{\star^2}} & \sum_{j=1}^{|\mathbf{Y}_{i,t}|}\left(\frac{\mathbf{u}_{j,1}^{(1)}}{\sigma_j^\star}\right)^2 \end{bmatrix} \tag{2.113}$$

where

$$D_1 \triangleq \frac{1}{2}\left(\sum_{j,k}^{|\mathbf{Y}i,t|}\frac{\|\mathbf{u}_{j,1}\otimes\mathbf{u}_{k,1}\|^2}{\sigma_j^{\star^2}\sigma_k^{\star^2}}\right) \tag{2.114}$$

Postmultiplying equation (2.113) by $\bar{\mathbf{G}}_1^\top$ we obtain a matrix with the elements,

$$\left[\left(\bar{\mathbf{G}}_1^\top \bar{\mathbf{G}}_1\right)^{-1}\bar{\mathbf{G}}_1^\top\right]_{1:2,m} = \frac{1}{D_1}\begin{bmatrix} \left(\frac{\mathbf{u}_{m,1}^{(1)}}{\sigma_m^\star}\right)\sum_{j=1}^{|\mathbf{Y}_{i,t}|}\left(\frac{\mathbf{u}_{j,1}^{(2)}}{\sigma_j^\star}\right)^2 - \left(\frac{\mathbf{u}_{m,1}^{(2)}}{\sigma_m^\star}\right)\sum_{j=1}^{|\mathbf{Y}_{i,t}|}\frac{\mathbf{u}_{j,1}^{(1)}\mathbf{u}_{j,1}^{(2)}}{\sigma_j^{\star^2}} \\ \left(\frac{\mathbf{u}_{m,1}^{(2)}}{\sigma_m^\star}\right)\sum_{j=1}^{|\mathbf{Y}_{i,t}|}\left(\frac{\mathbf{u}_{j,1}^{(1)}}{\sigma_j^\star}\right)^2 - \left(\frac{\mathbf{u}_{m,1}^{(1)}}{\sigma_m^\star}\right)\sum_{j=1}^{|\mathbf{Y}_{i,t}|}\frac{\mathbf{u}_{j,1}^{(1)}\mathbf{u}_{j,1}^{(2)}}{\sigma_j^{\star^2}} \end{bmatrix} \tag{2.115}$$

Further premultiplying by $\bar{\mathbf{G}}_1$ yields the elements of $\mathbf{P}_{G_1}$ as,

$$[\mathbf{P}_{G_1}]_{n,m} = \frac{1}{D_1}\Bigg(\left(\frac{\mathbf{u}_{n,1}^{(1)}}{\sigma_n^\star}\right)\left(\left(\frac{\mathbf{u}_{m,1}^{(1)}}{\sigma_m^\star}\right)\sum_{j=1}^{|\mathbf{Y}_{i,t}|}\left(\frac{\mathbf{u}_{j,1}^{(2)}}{\sigma_j^\star}\right)^2 - \left(\frac{\mathbf{u}_{m,1}^{(2)}}{\sigma_m^\star}\right)\sum_{j=1}^{|\mathbf{Y}_{i,t}|}\frac{\mathbf{u}_{j,1}^{(1)}\mathbf{u}_{j,1}^{(2)}}{\sigma_j^{\star^2}}\right)$$

$$+\left(\frac{\mathbf{u}_{n,1}^{(2)}}{\sigma_n^\star}\right)\left(\left(\frac{\mathbf{u}_{m,1}^{(2)}}{\sigma_m^\star}\right)\sum_{j=1}^{|\mathbf{Y}_{i,t}|}\left(\frac{\mathbf{u}_{j,1}^{(1)}}{\sigma_j^\star}\right)^2 - \left(\frac{\mathbf{u}_{m,1}^{(1)}}{\sigma_m^\star}\right)\sum_{j=1}^{|\mathbf{Y}_{i,t}|}\frac{\mathbf{u}_{j,1}^{(1)}\mathbf{u}_{j,1}^{(2)}}{\sigma_j^{\star^2}}\right)\Bigg) \tag{2.116}$$

By straightforward algebraic manipulations this can be simplified as,

$$[\mathbf{P}_{G_1}]_{n,m} = \frac{1}{D_1}\sum_{j=1}^{|\mathbf{Y}_{i,t}|}\frac{\|\mathbf{u}_{n,1}\otimes\mathbf{u}_{j,1}\|\|\mathbf{u}_{m,1}\otimes\mathbf{u}_{j,1}\|}{\sigma_n^\star\sigma_m^\star\sigma_j^{\star^2}} \tag{2.117}$$

Subtracting the identity matrix by $\mathbf{P}_{G_1}$ and postmultiplying by $\bar{\mathbf{G}}_2$ yields the matrix with elements:

$$\left[\left(\mathbf{I}-\mathbf{P}_{G_1}\right)\bar{\mathbf{G}}_2\right]_{n,m} = \frac{D_1\left(\frac{\mathbf{u}_{n,2}^{(m)}}{\sigma_n^\star}\right) - \left(\sum_{j,k}^{|\mathbf{Y}_{i,t}|}\frac{\mathbf{u}_{k,2}^{(m)}\|\mathbf{u}_{n,1}\otimes\mathbf{u}_{j,1}\|\|\mathbf{u}_{k,1}\otimes\mathbf{u}_{j,1}\|}{\sigma_n^\star\sigma_k^{\star^2}\sigma_j^{\star^2}}\right)}{D_1} \tag{2.118}$$

Finally premultiplying (2.118) by $\bar{\mathbf{G}}_2^\top$ produces:

$$\left[\bar{\mathbf{G}}_2^\top \left(\mathbf{I} - \mathbf{P}_{G_1}\right) \bar{\mathbf{G}}_2\right]_{n,m} = \underbrace{\sum_{p=1}^{|\mathbf{Y}_{i,t}|} \left(\frac{\mathbf{u}_{p,2}^{(n)} \mathbf{u}_{p,2}^{(m)}}{\sigma_p^{\star^2}}\right)}_{S_{n,m,a}} - \underbrace{\frac{1}{D_1} \left(\sum_{j,k,p}^{|\mathbf{Y}_{i,t}|} \frac{\mathbf{u}_{p,2}^{(n)} \mathbf{u}_{k,2}^{(m)} \|\mathbf{u}_{p,1} \otimes \mathbf{u}_{j,1}\| \|\mathbf{u}_{k,1} \otimes \mathbf{u}_{j,1}\|}{\sigma_p^{\star^2} \sigma_k^{\star^2} \sigma_j^{\star^2}}\right)}_{S_{n,m,b}} \tag{2.119}$$

The determinant of this $2 \times 2$ matrix can then be written as:

$$\begin{aligned} &\det\left(\left[\bar{\mathbf{G}}_2^\top \left(\mathbf{I} - \mathbf{P}_{\mathrm{G}_1}\right) \bar{\mathbf{G}}_2\right]\right) = \left(\mathrm{S}_{1,1,\mathrm{a}} - \mathrm{S}_{1,1,\mathrm{b}}\right)\left(\mathrm{S}_{2,2,\mathrm{a}} - \mathrm{S}_{2,2,\mathrm{b}}\right) - \left(\mathrm{S}_{1,2,\mathrm{a}} - \mathrm{S}_{1,2,\mathrm{b}}\right)^2 \\ &= \underbrace{\left(S_{1,1,a} S_{2,2,a} - S_{1,2,a}^2\right)}_{L_1} - \underbrace{\left(S_{1,1,a} S_{2,2,b} + S_{2,2,a} S_{1,1,b} - 2 S_{1,2,a} S_{1,2,b}\right)}_{L_2} + \underbrace{\left(S_{1,1,b} S_{2,2,b} - S_{1,2,b}^2\right)}_{L_3} \end{aligned} \tag{2.120}$$

Evaluating the first two terms in equation (2.120) we arrive at,

$$\begin{aligned} L_1 &= \left(\sum_{p=1}^{|\mathbf{Y}_{i,t}|} \left(\frac{\mathbf{u}_{p,2}^{(1)}}{\sigma_p^\star}\right)^2\right)\left(\sum_{p=1}^{|\mathbf{Y}_{i,t}|} \left(\frac{\mathbf{u}_{p,2}^{(2)}}{\sigma_p^\star}\right)^2\right) - \left(\sum_{p=1}^{|\mathbf{Y}_{i,t}|} \left(\frac{\mathbf{u}_{p,2}^{(1)} \mathbf{u}_{p,2}^{(2)}}{\sigma_p^{\star^2}}\right)\right)^2 \\ &= \frac{1}{2} \sum_{j,k}^{|\mathbf{Y}_{i,t}|} \frac{\|\mathbf{u}_{j,2} \otimes \mathbf{u}_{k,2}\|^2}{\sigma_j^{\star^2} \sigma_k^{\star^2}} \triangleq D_2 \end{aligned} \tag{2.121}$$

$$L_2 = \frac{1}{D_1} \sum_{j,k,p,l}^{|\mathbf{Y}_{i,t}|} \frac{\|\mathbf{u}_{p,1} \otimes \mathbf{u}_{j,1}\| \|\mathbf{u}_{k,1} \otimes \mathbf{u}_{j,1}\|}{\sigma_p^{\star^2} \sigma_k^{\star^2} \sigma_j^{\star^2} \sigma_l^{\star^2}} \underbrace{\left(\mathbf{u}_{p,2}^{(2)} \mathbf{u}_{k,2}^{(2)} \mathbf{u}_{l,2}^{(1)^2} + \mathbf{u}_{p,2}^{(1)} \mathbf{u}_{k,2}^{(1)} \mathbf{u}_{l,2}^{(2)^2} - 2 \mathbf{u}_{p,2}^{(1)} \mathbf{u}_{k,2}^{(2)} \mathbf{u}_{l,2}^{(1)} \mathbf{u}_{l,2}^{(2)}\right)}_{L_2^T} \tag{2.122}$$

The term $L_2^T$ in (2.122) can be simplified to,

$$\begin{aligned} L_2^T &= \mathbf{u}_{p,2}^{(1)} \mathbf{u}_{l,2}^{(2)} \left(\mathbf{u}_{k,2}^{(1)} \mathbf{u}_{l,2}^{(2)} - \mathbf{u}_{k,2}^{(2)} \mathbf{u}_{l,2}^{(1)}\right) + \mathbf{u}_{l,2}^{(1)} \mathbf{u}_{k,2}^{(2)} \left(\mathbf{u}_{p,2}^{(2)} \mathbf{u}_{l,2}^{(1)} - \mathbf{u}_{p,2}^{(1)} \mathbf{u}_{l,2}^{(2)}\right) \\ &= \mathbf{u}_{p,2}^{(1)} \mathbf{u}_{l,2}^{(2)} \|\mathbf{u}_{k,2} \otimes \mathbf{u}_{l,2}\| + \mathbf{u}_{l,2}^{(1)} \mathbf{u}_{k,2}^{(2)} \|\mathbf{u}_{l,2} \otimes \mathbf{u}_{p,2}\| \end{aligned} \tag{2.123}$$

Substituting (2.123) back into (2.122) we can express $L_2$ as,

$$\begin{aligned} L_2 = &\underbrace{\frac{1}{D_1} \sum_{j,k,p,l}^{|\mathbf{Y}_{i,t}|} \frac{\|\mathbf{u}_{p,1} \otimes \mathbf{u}_{j,1}\| \|\mathbf{u}_{k,1} \otimes \mathbf{u}_{j,1}\| \|\mathbf{u}_{k,2} \otimes \mathbf{u}_{l,2}\| \mathbf{u}_{p,2}^{(1)} \mathbf{u}_{l,2}^{(2)}}{\sigma_p^{\star^2} \sigma_k^{\star^2} \sigma_j^{\star^2} \sigma_l^{\star^2}}}_{L_{2,1}} \\ &+ \frac{1}{D_1} \sum_{j,k,p,l}^{|\mathbf{Y}_{i,t}|} \frac{\|\mathbf{u}_{p,1} \otimes \mathbf{u}_{j,1}\| \|\mathbf{u}_{k,1} \otimes \mathbf{u}_{j,1}\| \|\mathbf{u}_{l,2} \otimes \mathbf{u}_{p,2}\| \mathbf{u}_{l,2}^{(1)} \mathbf{u}_{k,2}^{(2)}}{\sigma_p^{\star^2} \sigma_k^{\star^2} \sigma_j^{\star^2} \sigma_l^{\star^2}} \end{aligned} \tag{2.124}$$

Splitting the sum in this fashion allows us to recognize that the first term, $L_{2,1}$ can be rewritten using the mapping $k \to p', p \to k'$:

$$L_{2,1} = \sum_{j,k',p',l}^{|\mathbf{Y}_{i,t}|} \frac{\|\mathbf{u}_{k',1} \otimes \mathbf{u}_{j,1}\|\|\mathbf{u}_{p',1} \otimes \mathbf{u}_{j,1}\|\|\mathbf{u}_{p',2} \otimes \mathbf{u}_{l,2}\|\mathbf{u}_{k',2}^{(1)}\mathbf{u}_{l,2}^{(2)}}{\sigma_{k'}^{\star^2}\sigma_{p'}^{\star^2}\sigma_{j}^{\star^2}\sigma_{l}^{\star^2}} \tag{2.125}$$

This allows a final simplification of (2.124) to:

$$L_2 = \frac{1}{D_1}\sum_{j,k,p,l}^{|\mathbf{Y}_{i,t}|} \frac{\|\mathbf{u}_{p,1} \otimes \mathbf{u}_{j,1}\|\|\mathbf{u}_{k,1} \otimes \mathbf{u}_{j,1}\|\|\mathbf{u}_{l,2} \otimes \mathbf{u}_{p,2}\|\|\mathbf{u}_{l,2} \otimes \mathbf{u}_{k,2}\|}{\sigma_{p}^{\star^2}\sigma_{k}^{\star^2}\sigma_{j}^{\star^2}\sigma_{l}^{\star^2}} \tag{2.126}$$

The last term in (2.120) involves a product of two triple sums, meaning that one can write:

$$\begin{aligned} L_3 &= \frac{1}{D_1^2}\sum_{j,n,k,m,l,p}^{|\mathbf{Y}_{i,t}|}\Bigg(\frac{\|\mathbf{u}_{p,1} \otimes \mathbf{u}_{j,1}\|\|\mathbf{u}_{k,1} \otimes \mathbf{u}_{j,1}\|\|\mathbf{u}_{l,1} \otimes \mathbf{u}_{m,1}\|\|\mathbf{u}_{n,1} \otimes \mathbf{u}_{m,1}\|}{\sigma_{j}^{\star^2}\sigma_{n}^{\star^2}\sigma_{k}^{\star^2}\sigma_{m}^{\star^2}\sigma_{l}^{\star^2}\sigma_{p}^{\star^2}} \\ &\qquad \times \left(\mathbf{u}_{k,2}^{(1)}\mathbf{u}_{p,2}^{(1)}\mathbf{u}_{l,2}^{(2)}\mathbf{u}_{n,2}^{(2)} - \mathbf{u}_{k,2}^{(1)}\mathbf{u}_{p,2}^{(2)}\mathbf{u}_{l,2}^{(1)}\mathbf{u}_{n,2}^{(2)}\right)\Bigg) \\ &= \frac{1}{D_1^2}\sum_{j,n,k,m,l,p}^{|\mathbf{Y}_{i,t}|} \frac{\|\mathbf{u}_{p,1} \otimes \mathbf{u}_{j,1}\|\|\mathbf{u}_{k,1} \otimes \mathbf{u}_{j,1}\|\|\mathbf{u}_{l,1} \otimes \mathbf{u}_{m,1}\|\|\mathbf{u}_{n,1} \otimes \mathbf{u}_{m,1}\|\|\mathbf{u}_{p,2} \otimes \mathbf{u}_{l,2}\|\mathbf{u}_{n,2}^{(2)}\mathbf{u}_{k,2}^{(1)}}{\sigma_{j}^{\star^2}\sigma_{n}^{\star^2}\sigma_{k}^{\star^2}\sigma_{m}^{\star^2}\sigma_{l}^{\star^2}\sigma_{p}^{\star^2}} \end{aligned} \tag{2.127}$$

Splitting the sum in (2.127) in half, with one term invoking the mapping $l \to p', p \to l'$ yields the expression,

$$\begin{aligned} L_3 = \frac{1}{2D_1^2}\sum_{j,n,k,m,l,p}^{|\mathbf{Y}_{i,t}|}\Bigg(&\frac{\|\mathbf{u}_{k,1} \otimes \mathbf{u}_{j,1}\|\|\mathbf{u}_{n,1} \otimes \mathbf{u}_{m,1}\|\|\mathbf{u}_{p,2} \otimes \mathbf{u}_{l,2}\|\mathbf{u}_{n,2}^{(2)}\mathbf{u}_{k,2}^{(1)}}{\sigma_{j}^{\star^2}\sigma_{n}^{\star^2}\sigma_{k}^{\star^2}\sigma_{m}^{\star^2}\sigma_{l}^{\star^2}\sigma_{p}^{\star^2}} \\ &\times \underbrace{\left(\|\mathbf{u}_{p,1} \otimes \mathbf{u}_{j,1}\|\|\mathbf{u}_{l,1} \otimes \mathbf{u}_{m,1}\| - \|\mathbf{u}_{p,1} \otimes \mathbf{u}_{m,1}\|\|\mathbf{u}_{l,1} \otimes \mathbf{u}_{j,1}\|\right)}_{L_3^T}\Bigg) \end{aligned} \tag{2.128}$$

An identity regarding cross products is now derived that will allow $L_3^T$ to be simplified. Namely, we have that:

$$\|\mathbf{u}_{1,i} \otimes \mathbf{u}_{2,i}\|\|\mathbf{u}_{3,i} \otimes \mathbf{u}_{4,i}\| - \|\mathbf{u}_{1,i} \otimes \mathbf{u}_{4,i}\|\|\mathbf{u}_{3,i} \otimes \mathbf{u}_{2,i}\| = \|\mathbf{u}_{3,i} \otimes \mathbf{u}_{1,i}\|\|\mathbf{u}_{4,i} \otimes \mathbf{u}_{2,i}\| \tag{2.129}$$

To prove this is true, note that the left-hand side of (2.129) can be written as,

$$\begin{vmatrix} \mathbf{u}_{1,i}^{(1)} & \mathbf{u}_{2,i}^{(1)} \\ \mathbf{u}_{1,i}^{(2)} & \mathbf{u}_{2,i}^{(2)} \end{vmatrix} \begin{vmatrix} \mathbf{u}_{3,i}^{(1)} & \mathbf{u}_{4,i}^{(1)} \\ \mathbf{u}_{3,i}^{(2)} & \mathbf{u}_{4,i}^{(2)} \end{vmatrix} - \begin{vmatrix} \mathbf{u}_{1,i}^{(1)} & \mathbf{u}_{4,i}^{(1)} \\ \mathbf{u}_{1,i}^{(2)} & \mathbf{u}_{4,i}^{(2)} \end{vmatrix} \begin{vmatrix} \mathbf{u}_{3,i}^{(1)} & \mathbf{u}_{2,i}^{(1)} \\ \mathbf{u}_{3,i}^{(2)} & \mathbf{u}_{2,i}^{(2)} \end{vmatrix} = \left| \begin{bmatrix} \mathbf{u}_{1,i}^\top \\ \mathbf{u}_{2,i}^\top \end{bmatrix} \begin{bmatrix} \mathbf{u}_{3,i} & \mathbf{u}_{4,i} \end{bmatrix} \right| - \left| \begin{bmatrix} \mathbf{u}_{3,i}^\top \\ \mathbf{u}_{2,i}^\top \end{bmatrix} \begin{bmatrix} \mathbf{u}_{1,i} & \mathbf{u}_{4,i} \end{bmatrix} \right|$$
$$= \left( \left( \mathbf{u}_{1,i}^\top \mathbf{u}_{3,i} \right) \left( \mathbf{u}_{2,i}^\top \mathbf{u}_{4,i} \right) - \left( \mathbf{u}_{1,i}^\top \mathbf{u}_{4,i} \right) \left( \mathbf{u}_{2,i}^\top \mathbf{u}_{3,i} \right) \right) - \left( \left( \mathbf{u}_{3,i}^\top \mathbf{u}_{1,i} \right) \left( \mathbf{u}_{2,i}^\top \mathbf{u}_{4,i} \right) - \left( \mathbf{u}_{3,i}^\top \mathbf{u}_{4,i} \right) \left( \mathbf{u}_{2,i}^\top \mathbf{u}_{1,i} \right) \right)$$
$$= \mathbf{u}_{2,i}^\top \left( \mathbf{u}_{1,i} \mathbf{u}_{3,i}^\top - \mathbf{u}_{3,i} \mathbf{u}_{1,i}^\top \right) \mathbf{u}_{4,i} = \mathbf{u}_{2,i}^\top \left( \begin{bmatrix} \mathbf{u}_{1,i}^{(1)} \mathbf{u}_{3,i}^{(1)} & \mathbf{u}_{1,i}^{(1)} \mathbf{u}_{3,i}^{(2)} \\ \mathbf{u}_{1,i}^{(2)} \mathbf{u}_{3,i}^{(1)} & \mathbf{u}_{1,i}^{(2)} \mathbf{u}_{3,i}^{(2)} \end{bmatrix} - \begin{bmatrix} \mathbf{u}_{1,i}^{(1)} \mathbf{u}_{3,i}^{(1)} & \mathbf{u}_{1,i}^{(2)} \mathbf{u}_{3,i}^{(1)} \\ \mathbf{u}_{1,i}^{(1)} \mathbf{u}_{3,i}^{(2)} & \mathbf{u}_{1,i}^{(2)} \mathbf{u}_{3,i}^{(2)} \end{bmatrix} \right) \mathbf{u}_{4,i}$$
$$= \mathbf{u}_{2,i}^\top \begin{bmatrix} 0 & -\|\mathbf{u}_{3,i} \otimes \mathbf{u}_{1,i}\| \\ \|\mathbf{u}_{3,i} \otimes \mathbf{u}_{1,i}\| & 0 \end{bmatrix} \mathbf{u}_{4,i} = \begin{bmatrix} \mathbf{u}_{2,i}^{(1)} & \mathbf{u}_{2,i}^{(2)} \end{bmatrix} \begin{bmatrix} -\mathbf{u}_{4,i}^{(2)} \|\mathbf{u}_{3,i} \otimes \mathbf{u}_{1,i}\| \\ \mathbf{u}_{4,i}^{(1)} \|\mathbf{u}_{3,i} \otimes \mathbf{u}_{1,i}\| \end{bmatrix}$$
$$= \|\mathbf{u}_{3,i} \otimes \mathbf{u}_{1,i}\| \|\mathbf{u}_{4,i} \otimes \mathbf{u}_{2,i}\| \quad (2.130)$$

Applying this identity to $L_3^T$ in (2.128), then repeating the same sum-splitting procedure this time with the mapping $j \to m', m \to j'$ allows for the following steps:

$$L_3 = \frac{1}{2D_1^2} \sum_{j,n,k,m,l,p}^{|\mathbf{Y}_{i,t}|} \frac{\|\mathbf{u}_{k,1} \otimes \mathbf{u}_{j,1}\| \|\mathbf{u}_{n,1} \otimes \mathbf{u}_{m,1}\| \|\mathbf{u}_{p,2} \otimes \mathbf{u}_{l,2}\| \|\mathbf{u}_{l,1} \otimes \mathbf{u}_{p,1}\| \|\mathbf{u}_{m,1} \otimes \mathbf{u}_{j,1}\| \mathbf{u}_{n,2}^{(2)} \mathbf{u}_{k,2}^{(1)}}{\sigma_j^{\star 2} \sigma_n^{\star 2} \sigma_k^{\star 2} \sigma_m^{\star 2} \sigma_l^{\star 2} \sigma_p^{\star 2}}$$
$$= \frac{1}{4D_1^2} \sum_{j,n,k,m,l,p}^{|\mathbf{Y}_{i,t}|} \left( \frac{\|\mathbf{u}_{p,2} \otimes \mathbf{u}_{l,2}\| \|\mathbf{u}_{l,1} \otimes \mathbf{u}_{p,1}\| \|\mathbf{u}_{m,1} \otimes \mathbf{u}_{j,1}\| \mathbf{u}_{n,2}^{(2)} \mathbf{u}_{k,2}^{(1)}}{\sigma_j^{\star 2} \sigma_n^{\star 2} \sigma_k^{\star 2} \sigma_m^{\star 2} \sigma_l^{\star 2} \sigma_p^{\star 2}} \right.$$
$$\left. \times \left( \|\mathbf{u}_{k,1} \otimes \mathbf{u}_{j,1}\| \|\mathbf{u}_{n,1} \otimes \mathbf{u}_{m,1}\| - \|\mathbf{u}_{k,1} \otimes \mathbf{u}_{m,1}\| \|\mathbf{u}_{n,1} \otimes \mathbf{u}_{j,1}\| \right) \right)$$
$$= \frac{1}{4D_1^2} \sum_{j,n,k,m,l,p}^{|\mathbf{Y}_{i,t}|} \frac{\|\mathbf{u}_{p,2} \otimes \mathbf{u}_{l,2}\| \|\mathbf{u}_{l,1} \otimes \mathbf{u}_{p,1}\| \|\mathbf{u}_{m,1} \otimes \mathbf{u}_{j,1}\|^2 \|\mathbf{u}_{n,1} \otimes \mathbf{u}_{k,1}\| \mathbf{u}_{n,2}^{(2)} \mathbf{u}_{k,2}^{(1)}}{\sigma_j^{\star 2} \sigma_n^{\star 2} \sigma_k^{\star 2} \sigma_m^{\star 2} \sigma_l^{\star 2} \sigma_p^{\star 2}}$$
$$= \frac{1}{2D_1} \sum_{n,k,l,p}^{|\mathbf{Y}_{i,t}|} \frac{\|\mathbf{u}_{p,2} \otimes \mathbf{u}_{l,2}\| \|\mathbf{u}_{l,1} \otimes \mathbf{u}_{p,1}\| \|\mathbf{u}_{n,1} \otimes \mathbf{u}_{k,1}\| \mathbf{u}_{n,2}^{(2)} \mathbf{u}_{k,2}^{(1)}}{\sigma_n^{\star 2} \sigma_k^{\star 2} \sigma_l^{\star 2} \sigma_p^{\star 2}} \quad (2.131)$$

We are now in a position to complete the final simplification to $L_3$, namely by once more splitting the sum and applying the mapping $k \to n', n \to k'$ to one of the terms, yields:

$$L_3 = \frac{1}{4D_1} \sum_{n,k,l,p}^{|\mathbf{Y}_{i,t}|} \frac{\|\mathbf{u}_{p,2} \otimes \mathbf{u}_{l,2}\| \|\mathbf{u}_{l,1} \otimes \mathbf{u}_{p,1}\| \|\mathbf{u}_{n,1} \otimes \mathbf{u}_{k,1}\| \mathbf{u}_{n,2}^{(2)} \mathbf{u}_{k,2}^{(1)}}{\sigma_n^{\star 2} \sigma_k^{\star 2} \sigma_l^{\star 2} \sigma_p^{\star 2}} \left( \mathbf{u}_{n,2}^{(2)} \mathbf{u}_{k,2}^{(1)} - \mathbf{u}_{k,2}^{(2)} \mathbf{u}_{n,2}^{(1)} \right)$$
$$= \frac{1}{4D_1} \sum_{n,k,l,p}^{|\mathbf{Y}_{i,t}|} \frac{\|\mathbf{u}_{l,1} \otimes \mathbf{u}_{p,1}\| \|\mathbf{u}_{l,2} \otimes \mathbf{u}_{p,2}\| \|\mathbf{u}_{n,1} \otimes \mathbf{u}_{k,1}\| \|\mathbf{u}_{n,2} \otimes \mathbf{u}_{k,2}\|}{\sigma_n^{\star 2} \sigma_k^{\star 2} \sigma_l^{\star 2} \sigma_p^{\star 2}}$$
$$= \frac{1}{4D_1} \left( \sum_{j,k}^{|\mathbf{Y}_{i,t}|} \frac{\|\mathbf{u}_{j,1} \otimes \mathbf{u}_{k,1}\| \|\mathbf{u}_{j,2} \otimes \mathbf{u}_{k,2}\|}{\sigma_j^{\star 2} \sigma_k^{\star 2}} \right)^2 \quad (2.132)$$

Combining equations (2.121), (2.126), and (2.132) yields for the determinant,

$$\det\left(\left[\bar{\mathbf{G}}_2^{\top}\left(\mathbf{I}-\mathbf{P}_{\mathrm{G}_1}\right)\bar{\mathbf{G}}_2\right]\right)=\mathrm{D}_2-\frac{1}{\mathrm{D}_1}\sum_{\mathrm{j,k,p,l}}^{|\mathbf{Y}_{\mathrm{i,t}}|}\frac{\|\mathbf{u}_{\mathrm{p},1}\otimes\mathbf{u}_{\mathrm{j},1}\|\|\mathbf{u}_{\mathrm{k},1}\otimes\mathbf{u}_{\mathrm{j},1}\|\|\mathbf{u}_{\mathrm{p},2}\otimes\mathbf{u}_{\mathrm{l},2}\|\|\mathbf{u}_{\mathrm{k},2}\otimes\mathbf{u}_{\mathrm{l},2}\|}{\sigma_{\mathrm{p}}^{\star^2}\sigma_{\mathrm{k}}^{\star^2}\sigma_{\mathrm{j}}^{\star^2}\sigma_{\mathrm{l}}^{\star^2}}+\frac{1}{4D_1}\left(\sum_{j,k}^{|\mathbf{Y}_{i,t}|}\frac{\|\mathbf{u}_{j,1}\otimes\mathbf{u}_{k,1}\|\|\mathbf{u}_{j,2}\otimes\mathbf{u}_{k,2}\|}{\sigma_j^{\star^2}\sigma_k^{\star^2}}\right)^2 \tag{2.133}$$

The LDL decomposition allows us to use this result to obtain $\det\left(\mathbf{J}\left(\mathbf{X}_{\mathbf{S}_{\mathrm{i,t}}}^{1:2}|\psi\left(\mathbf{X}_{\mathbf{F}_{\mathrm{i,t-1}}}\right)\right)\right)$ which is simply the product of (2.133) and $D_1$. The preceding expression is in fact of substantial value as it allows the Fast-Piecewise algorithm specifically developed for $|\mathbf{S}_{i,t}|=1$ to be leveraged. In fact, a recursive form for (2.133) has been developed that yields highly efficient computation and dramatic improvement over the employment of an optimization method such as GA. The implementation details and performance results will not be mentioned here but are planned for submission as a future publication. What follows instead is an exploratory consideration of the previous results that does not make use of Fast-Piecewise but allows one to formulate a reasonable configuration based on qualitative discussion.

It can be shown that (2.133) vanishes for $K<4$ meaning intuitively that two targets cannot be tracked with than less 4 sensors. Resting on the observation that this expression is a sum of cross-products between target-sensor vector pairs, one possibility is to split the $K$ sensors into two groups, each optimized for one of the targets. This intuitively works well for larger separations [1] as sensors close to one target will be largely unaffected by the influence from the other; it can be shown that (2.133) essentially reduces to the sum of (2.78) for both targets (with each sensor in the appropriate summation depending on which target it's centered about) as $\|\mathbf{x}_{1,t}-\mathbf{x}_{2,t}\|\to\infty$. It is not immediately obvious if a better choice exists for when the targets are close, particularly in the case where $\|\mathbf{x}_{1,t}-\mathbf{x}_{2,t}\|\leq r$, i.e., when the $r$-circles from each respective target overlap.

To get a better sense of reasonable configurations in this case, numerical optimization using Genetic Algorithm (with solutions again constrained only to maintain sensor-target separation $\geq r$) was conducted for four different target separations with a fixed sensor radius of $r=3$ for $K=4$ sensors. The results are shown in figure 2.9 for the case where there are no external interferers; filled dots within the centers of the plotted circles demarcate target locations, while different symbols correspond to sensor locations within a given computed configuration. Interestingly, one can observe that each configuration follows roughly the solution described above, regardless of target separation; every two out of four sensors lie on a given target's $r$ circle and the two vectors formed by said target with each of these sensors are approximately orthogonal. We can in fact develop an analytic solution

[1] Recalling the constraint that all sensors must be spaced at least $r$ from all targets.

which follows this configuration format and is guaranteed to satisfy the sensor-target separation constraint at least in the case of $K = 2$ sensors *for any target separation* as follows:

$$\begin{aligned}
\mathbf{q}_{1,t} &= \mathbf{x}_{1,t} + r\left[\cos\left(\angle_{1,t,c}\right) \quad \sin\left(\angle_{1,t,c}\right)\right] \\
\mathbf{q}_{2,t} &= \mathbf{x}_{1,t} + r\left[\cos\left(\angle_{1,t,c} + \tfrac{\pi}{2}\right) \quad \sin\left(\angle_{1,t,c} + \tfrac{\pi}{2}\right)\right] \\
\mathbf{q}_{3,t} &= \mathbf{x}_{2,t} + r\left[\cos\left(\angle_{2,t,c} - \tfrac{\pi}{2}\right) \quad \sin\left(\angle_{2,t,c} - \tfrac{\pi}{2}\right)\right] \\
\mathbf{q}_{4,t} &= \mathbf{x}_{2,t} + r\left[\cos\left(\angle_{2,t,c} - \pi\right) \quad \sin\left(\angle_{2,t,c} - \pi\right)\right]
\end{aligned} \tag{2.134}$$

where $\angle_{i,t,c}$ is the angle $\operatorname{atan2}\left(z_t - \mathbf{x}_{i,t}\right)$ modulo $2\pi$ between $\mathbf{x}_{i,t}$ and the "positive" intersection point, $z_t$, of the two $r$-circles centered about $\mathbf{x}_{1,t}$ and $\mathbf{x}_{2,t}$. The positive point meaning that which is oriented at a positive angle with respect to the line joining the two targets (and starting at $\mathbf{x}_{1,t}$).

One final possibility for two targets is to again split the total number of sensors into two groups and position them in the same fashion as in (2.100). In other words, when positioning a group of sensors corresponding to one target, the other target is treated as if it were interference. While this may not be optimal as the targets approach one another, it does allow the other approaches outlined for a single target to be trivially employed in this case and becomes increasingly significant as the total number of targets in the environment grows.

## 2.8 Performance Results

The results presented here aim to detail computer simulation results for each portion of MASCOT separately, then as a whole.

### 2.8.1 Agent Cooperation

In this section, two example scenarios are presented illustrating the performance of MASCOT agent cooperation as outlined in 2.6 and a comparison is made between "probabilistic information" information exchange (subsequently labeled COOP) and simple point-estimate exchange (labeled BASIC) in the computation of the particle weights according to (2.37). In either scenario, agents implement a partial version of the algorithm described in 2.2; there is no adapative partitioning here and each agent is simply allocated a single TOI to estimate. For this reason, a simpler notation than in section 2.6 will be adopted here regarding target locations. Namely, the $l$-th target state is denoted $\mathbf{x}_{l,t}$ and estimated by agent $l$; this can be transposed back to section 2.6 by simply writing that $\mathbf{X}_{\mathbf{S}_{l,t}} = \{\mathbf{x}_{l,t}\}$. Also, sensors are not positioned optimally here; they are simply uniformly spaced on a circle of radius $r$ centered about the target.

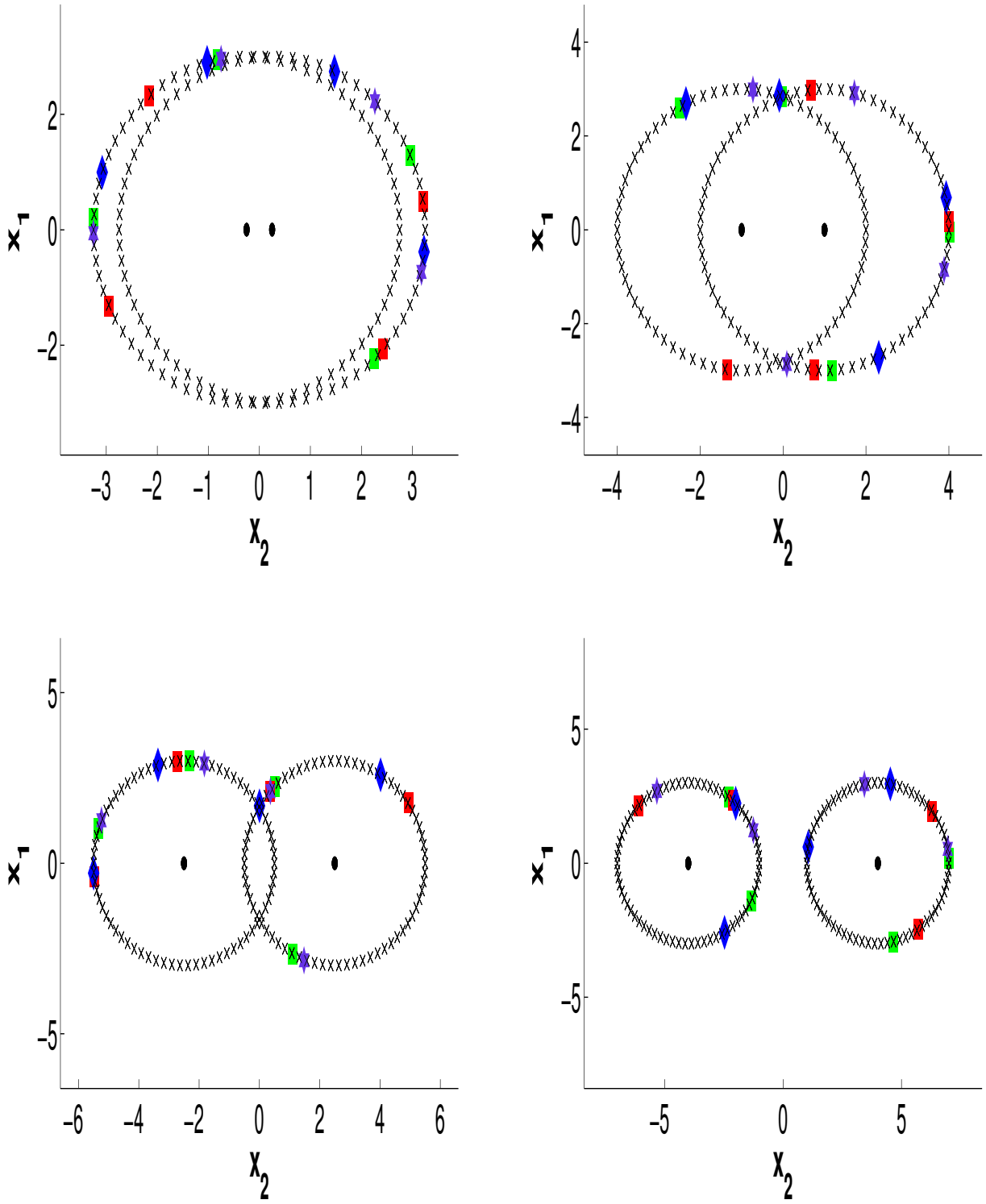


**Figure 2.9:** Optimal sensor configurations for two targets computed with Genetic Algorithm.

Specific trajectories for the targets were chosen to allow for controllable variation of their loca-

tions. The first scenario consists of two targets with dynamics modeled by

$$\begin{cases} \mathbf{x}_{l,t} = \mathbf{G}_l \mathbf{x}_{l,t-1}, & \text{if } \omega t + \pi \leq \frac{3\pi}{2} \\ \mathbf{x}_{l,t} = \mathbf{A}\mathbf{x}_{l,t-1} + \mathbf{v}_{l,t}, & \text{if } \omega t + \pi > \frac{3\pi}{2}, \end{cases} \tag{2.135}$$

with,

$$\mathbf{G}_l = \begin{bmatrix} 1 & 0 & \cos(\omega) & (-1)^l \sin(\omega) \\ 0 & 1 & (-1)^{l+1} \sin(\omega) & \cos(\omega) \\ 0 & 0 & \cos(\omega) & (-1)^l \sin(\omega) \\ 0 & 0 & (-1)^{l+1} \sin(\omega) & \cos(\omega) \end{bmatrix}, \tag{2.136}$$

and,

$$\mathbf{x}_{l,1} = \begin{bmatrix} -d & (-1)^{l+1} d & -\sqrt{\lambda}\omega \sin(\omega) & -\sqrt{\lambda}\omega \cos(\omega) \end{bmatrix}^\top. \tag{2.137}$$

In words, the two targets were initially separated by an amount $d$ and each followed a turning maneuver which caused them to converge at a final separation that was governed by the parameter $\lambda$. Upon completion of the maneuver, the targets switched to a constant-velocity trajectory with process noise perturbations. Figure 2.10 illustrates sample trajectories generated with various values for $\lambda$. The process noise intensity $\sigma_w$ of $\mathbf{v}_{l,t}$ was fixed at 0.002 during the "free-motion" portion of each path. Also, the agents did not use prior knowledge about the specific maneuver considered here. They assumed that the target dynamics followed the model in (2.10) at all times. In order to gain better insight as to the performance benefit realized with the proposed method, the following metrics are defined,

$$\|\bar{\mathbf{X}}\| = \frac{1}{T}\sum_{t=1}^{T} \|\mathbf{x}_{1,t}^{1:2} - \mathbf{x}_{2,t}^{1:2}\|, \qquad \|\hat{\mathbf{X}}_l\| = \frac{1}{T}\sum_{t=1}^{T} \|\hat{\mathbf{x}}_{l,t}^{1:2} - \mathbf{x}_{l,t}^{1:2}\|. \tag{2.138}$$

The value $\|\bar{\mathbf{X}}\|$ is recognized as simply the mean over all simulation time of the distance between the two targets, while $\|\hat{\mathbf{X}}_l\|$ is the mean error of the $l$th target estimate. A lower value of $\|\bar{\mathbf{X}}\|$ corresponds to a more challenging scenario; the targets are closer together for a longer period of time and there is a higher degree of inter-target interference between sensor measurements.

The scenario was run for 4000 trials, each for a duration of $T = 200$ steps. In each trial a particular value of $\lambda$ was used, starting with $\lambda = 195$ and incrementing by 0.2 at every new trial. The parameters $d$ and $\omega$ were fixed at 15 and 0.01, respectively. Other fixed parameter values include $\sigma_u = 0.005$, $\sigma_v = 0.02$, $\alpha = 2$, $\epsilon = 0$, $\Phi = 10$, $r = 4$. The number of particles was set to $M = 300$, and each agent had $K = 4$ sensors. Particle resampling was done adaptively, i.e., resampling was initiated at time $t$ if the effective particle size fell below a fixed threshold. This condition is explicitly written as,

$$\left( \sum_{m=1}^{M} \left( w_{l,t}^{(m)} \right)^2 \right)^{-1} \leq \frac{M}{16}. \tag{2.139}$$

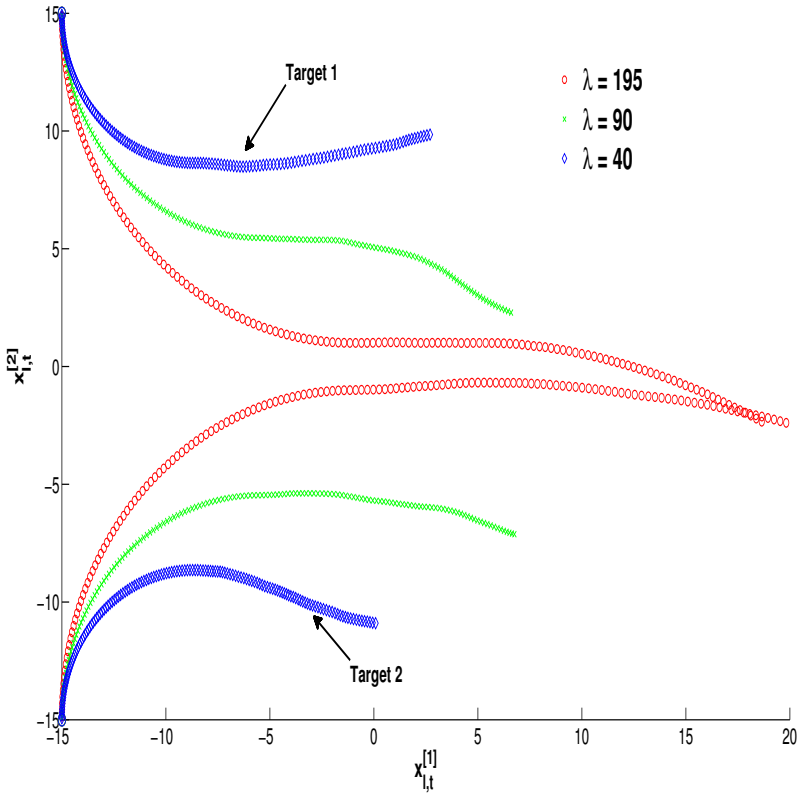


**Figure 2.10:** Sample trajectories of simulation scenario.

Note that, as mentioned previously, the issue of identity management between targets is considered a separate problem, is not addressed here, and may not be of concern depending on the application. The simulation trials where identity "loss" occurred were simply detected and discarded from the presented results.

In Fig. 2.11, we see a plot of the resulting $\|\bar{\mathbf{X}}\|$ against $\|\hat{\mathbf{X}}_l\|$ for each trial using BASIC and COOP (Proposed Method). One can immediately notice the superiority of the new method. There is a much larger sensitivity on $\|\bar{\mathbf{X}}\|$ in BASIC, whereas COOP only begins to deteriorate at very low values of $\|\bar{\mathbf{X}}\|$. When $\|\bar{\mathbf{X}}\|$ is small, a single target partition for each agent may be less appropriate; this would be when MASCOT (or even the symbiotic approach in [24] would merge two targets into a higher-dimensional filter.

The next scenario consisted of 20 targets with the following dynamics:

$$\begin{cases} \mathbf{x}_{l,t} = \mathrm{G}_l^{+}\mathbf{x}_{l,t-1}, & \text{if } 1 < l \leq 10 \text{ and } \omega t + \pi \leq \frac{3\pi}{2} \\ \mathbf{x}_{l,t} = \mathbf{A}\mathbf{x}_{l,t-1} + \mathbf{u}_{l,t}, & \text{if } 1 < l \leq 10 \text{ and } \omega t + \pi > \frac{3\pi}{2} \\ \mathbf{x}_{l,t} = \mathrm{G}_l^{-}\mathbf{x}_{l,t-1}, & \text{if } 11 < l \leq 20 \text{ and } \omega t + \pi \leq \frac{3\pi}{2} \\ \mathbf{x}_{l,t} = \mathbf{A}\mathbf{x}_{l,t-1} + \mathbf{u}_{l,t}, & \text{if } 11 < l \leq 20 \text{ and } \omega t + \pi > \frac{3\pi}{2} \\ \mathbf{x}_{l,t} = \mathbf{A}\mathbf{x}_{l,t-1} + \mathbf{u}_{l,t}, & \text{if } l = 1 \text{ or } l = 11 \end{cases} \tag{2.140}$$

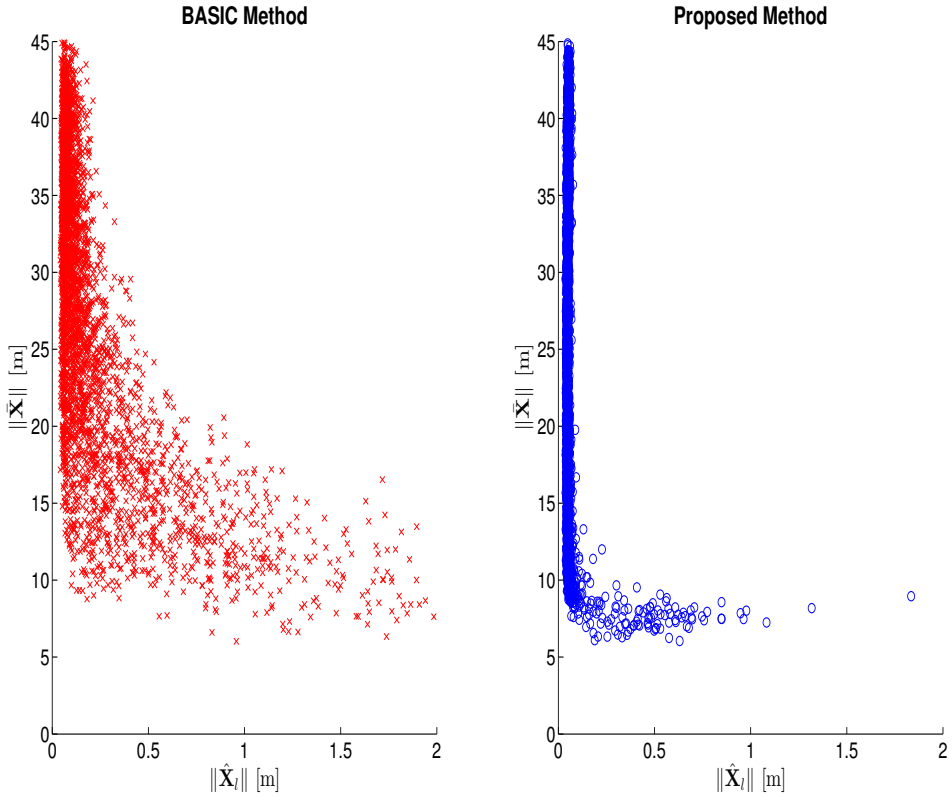


**Figure 2.11:** Error plot of the first simulation scenario.

with,

$$G_l^* = \begin{bmatrix} 1 & 0 & \cos(\omega) & -(*)\sin(\omega) \\ 0 & 1 & (*)\sin(\omega) & \cos(\omega) \\ 0 & 0 & \cos(\omega) & -(*)\sin(\omega) \\ 0 & 0 & (*)\sin(\omega) & \cos(\omega) \end{bmatrix}, \tag{2.141}$$

and the symbol $^*$ standing for $+$ or $-$,

$$\mathbf{x}_{l,1} = \begin{cases} \begin{bmatrix} -\frac{ld}{2} & \frac{ld}{2} & \sqrt{\lambda_l}\omega\sin(\omega) & -\sqrt{\lambda_l}\omega\cos(\omega) \end{bmatrix}^\top & \text{if } 1 < l \leq 10 \\ \begin{bmatrix} -\frac{ld}{2} & -\frac{ld}{2} & \sqrt{\lambda_l}\omega\sin(\omega) & \sqrt{\lambda_l}\omega\cos(\omega) \end{bmatrix}^\top & \text{if } 11 < l \leq 20 \\ \begin{bmatrix} -\frac{ld}{2} & \frac{ld}{2} & -v_0 & 0 \end{bmatrix}^\top & \text{if } l = 1 \\ \begin{bmatrix} -\frac{ld}{2} & -\frac{ld}{2} & -v_0 & 0 \end{bmatrix}^\top & \text{if } l = 11 \end{cases}, \tag{2.142}$$

where,

$$\begin{cases} \lambda_l = \varsigma^l \lambda_0, & \text{if } 1 \leq l \leq 10 \\ \lambda_l = \varsigma^{l-10} \lambda_0, & \text{if } 11 \leq l \leq 20 \end{cases}. \tag{2.143}$$

Again, this trajectory plan was chosen to allow for an initial separation between the targets. The separation was determined by the parameter $d$ and with a spatial convergence over time that was controlled by the parameters $\omega$, $\lambda_0$, and $\varsigma$.

In order to compare the performance of COOP with that of BASIC, a single fixed trajectory (shown in Fig. 2.12) was generated and used during the simulation runs, i.e., only the sensor measurements were randomly generated for each run. The parameters for this trajectory were set to $d = 22.0$, $\lambda_0 = 100.0$, $v_0 = 0.2$, and $\varsigma = 1.1$. The other simulation parameters were identical to the ones from the first scenario.

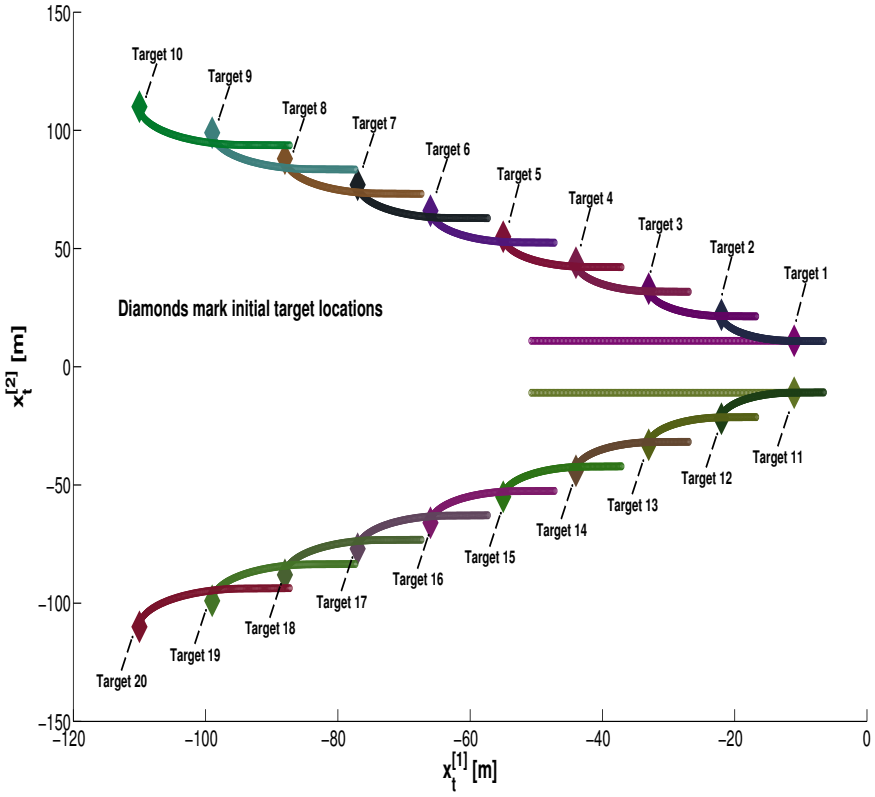


**Figure 2.12:** Target trajectory plot for the second scenario.

Note that targets 1 and 11 pose the greatest challenge since they maintain a relatively smaller distance from the largest number of targets. In Fig. 2.13 the "multi-target footprint" of target 1 is displayed in the considered trajectory. The top plot presents a count of how many of the other targets were within a given proximity from target 1 at that time. The bottom plot shows the "Signal-To-Total Measurement Ratio" (STMR) as a function of time. The STMR is defined as the ratio of the measurement contribution from target 1 (the signal) to the total measurement.

Figure 2.14 displays the MSE of target 1 computed over 50 trials for various values of $\sigma_v$ with BASIC and COOP. It can be seen that the error of BASIC sharply increases roughly around the time at which the STMR exhibits a sudden drop (corresponding to target 11 approaching within

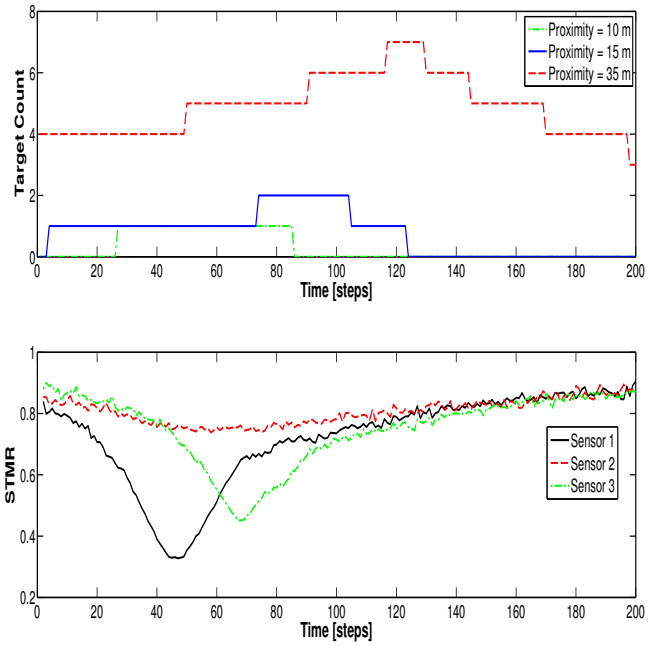


**Figure 2.13:** Cross-Target statistics for Target 1 in scenario 2.

close proximity of target 1). In all the cases, the performance of COOP is clearly superior to that of BASIC, and COOP experiences a nearly imperceptible increase in error for lower values of $\sigma_v$ when the value of STMR drops.

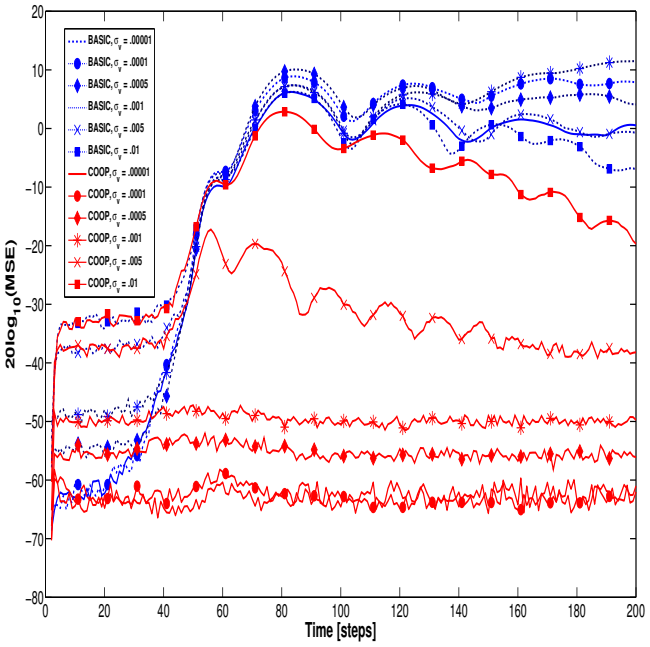


**Figure 2.14:** MSE of target 1 for various $\sigma_v$ in scenario 2.

### 2.8.2 Full MASCOT Implementation for Two Targets

The next scenario is used to illustrate the complete performance of the MASCOT algorithm, including agent cooperation, appropriate sensor positioning, and adaptive target space partitioning. Note in all subsequent results presented, it is assumed the sensors are positioned according to (2.100) whenever $|\mathbf{S}_{i,t}| \leq 2$.[1] This was chosen over other solutions presented due to its simplicity and allows key points of the adaptive partitioning to be illustrated clearly. The scenario consists of two targets initialized at an initial separation, which spatially converge as time progresses, and ultimately diverge. As in the previous scenarios, this trajectory is a forced maneuver primarily meant to demonstrate algorithm performance, however, the particle filter's do not have knowledge of the specific maneuver dynamics, i.e., the maneuvers are seen simply as a realization of the state process noise. The simulation was run with parameters set to $\Phi = 30$, $\alpha = 2$, $\epsilon = 0$, $\sigma_w = 0.08$, $\sigma_v = 0.04$, the total number of sensors $K = 8$, the target-sensor minimum separation $r = 2$, the number of particles per agent $M = 8000$, and resampling was done whenever the effective particle size is less than $M/4$. Shown in 2.15 are the target trajectories overlayed with the corresponding estimates; note that filled symbols mark the points at which the filter has chosen to jointly estimate $(\Theta_t^1)$ the two targets. In this case, joint estimation is chosen only when targets are within close proximity, while the partition is separated whenever they are further apart. While first intuition would be to simply establish the rule that partitions should merge whenever they are close enough together, this is not necessarily true. This can be seen in figure 2.16 where the log ratio of the objective function for the two possible partitions is plotted as a function of target separation for two different values of $\Phi$. It is clear that if $\Phi = 30$, the filter decides to conduct joint estimation when the targets are close enough, however when $\Phi = 10$ corresponding to a situation whereby each target interferes less with the others measurements, the filter does not conduct joint estimation at any separation.

To illustrate the benefit of MASCOT, the RMSE of each target location estimate was taken for 50 trials and compared to a filter which estimates both targets simultaneously, along with a filter which estimates each target individually (using agent-cooperation). The result is plotted in figure 2.17. It is interesting to note that joint estimation exhibits inferior performance at earlier times even though the earlier analysis in 2.6 suggested it would perform at least as well if not better than separated estimation at large target distances. This is a direct manifestation of the particle filter's deterioration due to dimensionality that is not captured in the error bound, yet is successfully avoided here through the regularization term $\mathbb{D}_M$. It can indeed be seen that MASCOT is able to reap the benefit of both possible partitions and avoid their respective weaknesses, thereby maximizing performance while minimizing computational demand.

[1] Although under the framework presented, it is possible for $|\mathbf{S}_{i,t}|$ to be any number, this is hard limited here to be at most 2.

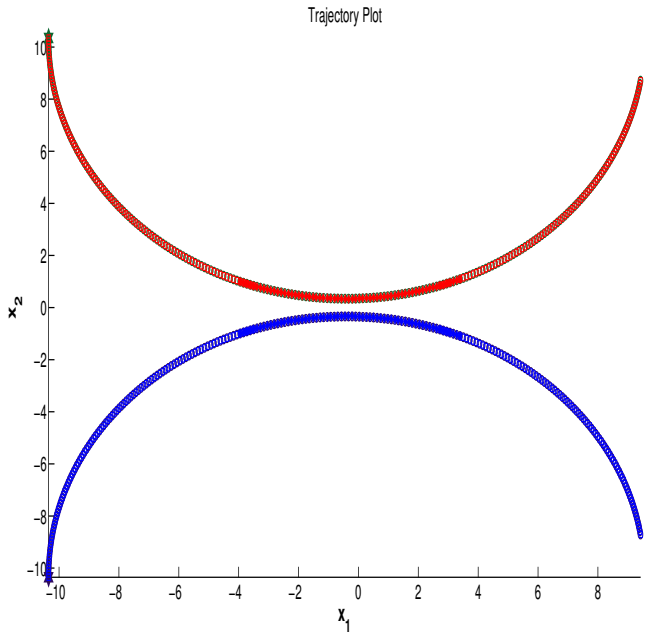


**Figure 2.15:** Two target and estimate trajectory plot for full MASCOT with $\Phi = 30$. Filled positions marks where joint estimation takes place.

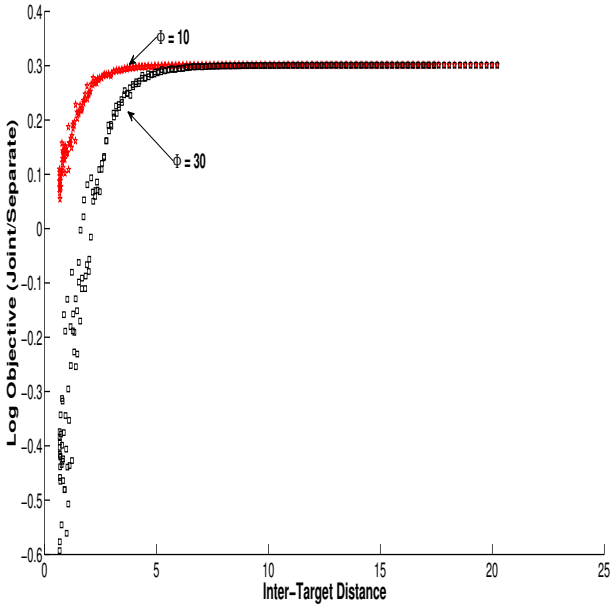


**Figure 2.16:** Log ratio of objective function for the two possible partitions when $\Phi = 10$ and $\Phi = 30$.

The final scenario illustrating MASCOT performance consists of six targets, with all parameters the same as the previous scenario, which follow the trajectory illustrated in 2.18. This scenario is considered relatively challenging due to the close proximity of multiple targets simultaneously, i.e., the interference component of each agent's measurements is significantly large. Figure 2.19 plots

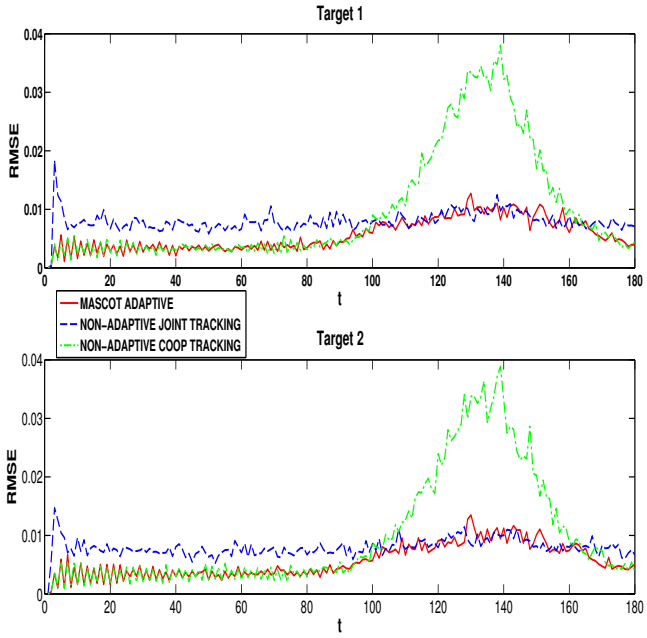


**Figure 2.17:** RMSE Performance Comparison between MASCOT adaptive filter and basic joint/separated estimation of both targets..

the distance from one target to another as a function of simulation time where we can see that this proximity is maintained for significant periods of time. The RMSE was again taken for 50 trials and is plotted for each target in figure 2.20; again the full Mascot filter is compared to the non-adaptive cooperative one[1]. While the non-adaptive filter does perform well, it is evident that significant error starts to occur at points of maximum interference. It is also immediately apparent that MASCOT is able to cope with these events resulting in the lower RMSE seen at each one. It is also important to note here that in this case, the sensor configuration employed is *far from optimal* for when MASCOT is employing a partition that consists of more than one target; sensors will position themselves as if the other target were interference while ignoring the true interfering sources. Essentially, this places a "handicap" on the best-attainable performance with MASCOT since it may choose a single-target partition in cases where there may be joint-target partitions that are substantially more favorable (yet which were not found under the basic sensor positioning method). As such, it is logical to expect a significant increase in performance by using the results mentioned in 2.7.2 to implement the Fast-Piecewise positioning method for joint-target estimation.

[1]Comparison to a joint filter was not done here since there is ambiguity as to which targets to pair together

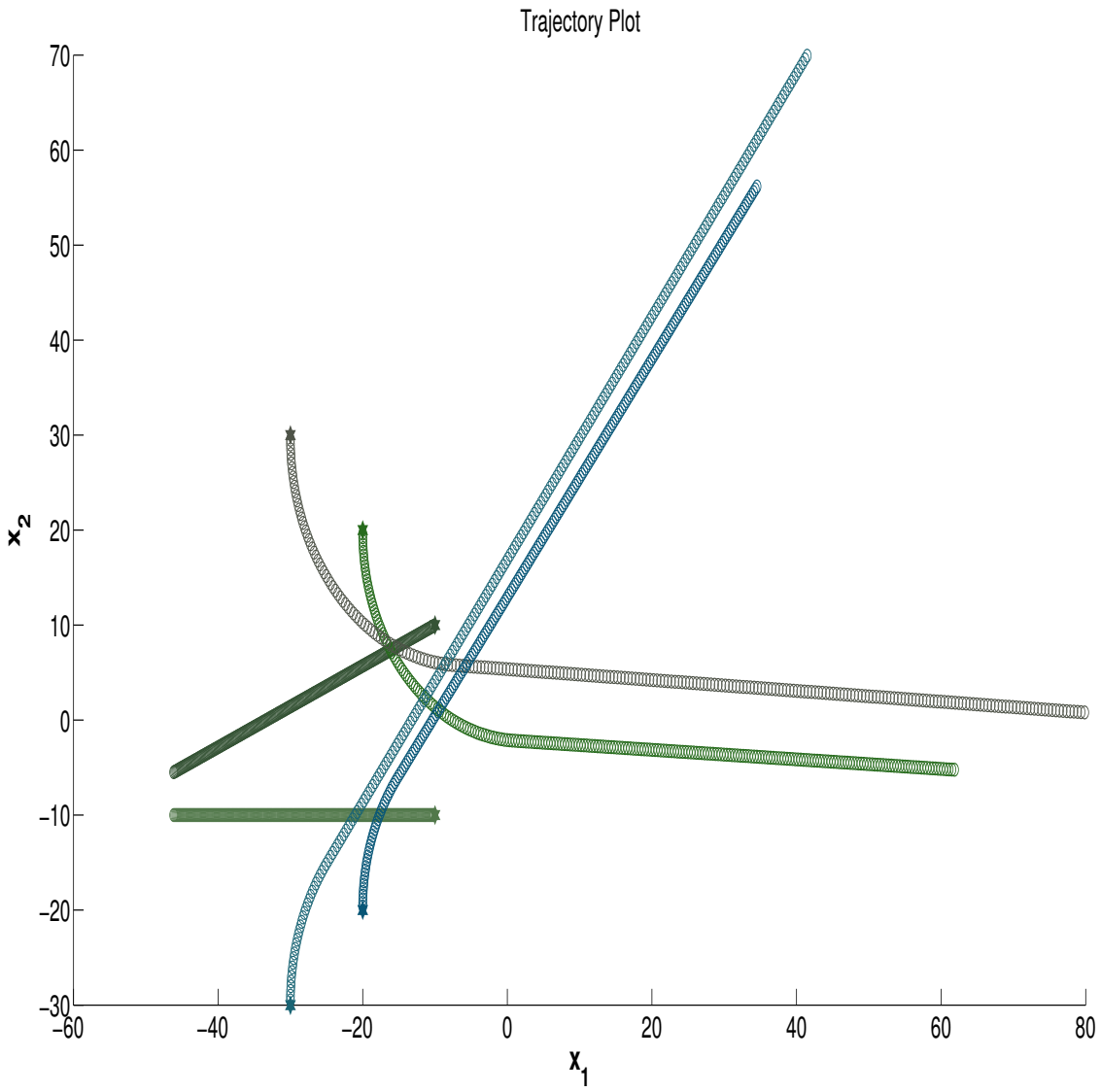


**Figure 2.18:** Six Target Trajectory for Full Mascot Simulation.

## 2.9 Concluding Remarks

In this chapter, the MASCOT framework has been outlined in generality and then fully specialized to a complete multi-agent system capable of tracking multiple targets using mobile RSSI sensors. Major features of the system addressed include agent cooperation via fusion of exchanged information, dynamic state space partitioning along with optimal sensor positioning in the presence of multiple localized interference sources (which is an important problem in its own rite). Performance results were demonstrated indicating the clear advantage of MASCOT in handling high-dimensional environments. Major areas that remain open are initialization and target detection. The way that MASCOT has been described here assumes that there is some initial time at which a fixed number

of targets were properly detected and a reasonably accurate initial estimate for the full target state $\mathbf{X}_t$ had been computed. It is always assumed that all targets within the environment that have a non-negligible effect on the sensor measurements are accounted for by some agent at all times; the presented implementation of MASCOT is not expected to perform well and will likely experience rapid performance deterioration if one or more interfering sources exist that have not been sufficiently localized by the algorithm, particularly at the time of target initialization. This is considered as external to the scope of MASCOT and focus is maintained on the situation where this ambiguity does not occur; if a target (or cluster of targets) exists within the environment, then it (or they) was being properly tracked in isolation (away from other existing targets) at some earlier point in time. Some work has been done as part of the dissertation in addressing unknown interfering sources [1] but incorporation into MASCOT was never fully attempted. Additionally, dynamic target detection (whereby the total target number is unknown, and targets can suddenly appear/disapper from the scene) was not considered as part of the thesis. It is believed that extension of the MASCOT algorithm using RFS-theoretic methods can be effective in handling this.

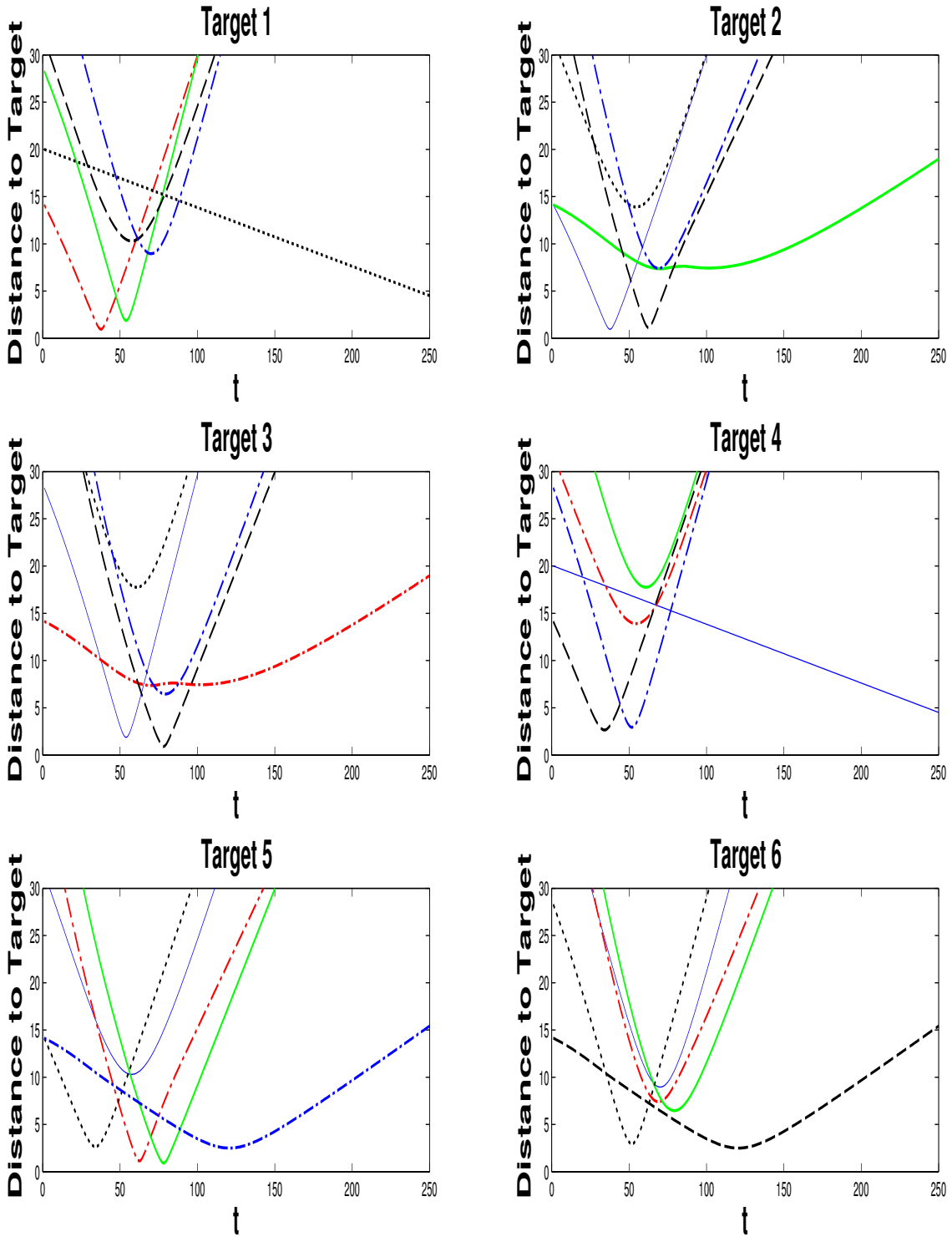


**Figure 2.19:** Inter-target distance as a function of time in the six-target scenario. Different patterned/colored lines each represent a different target.

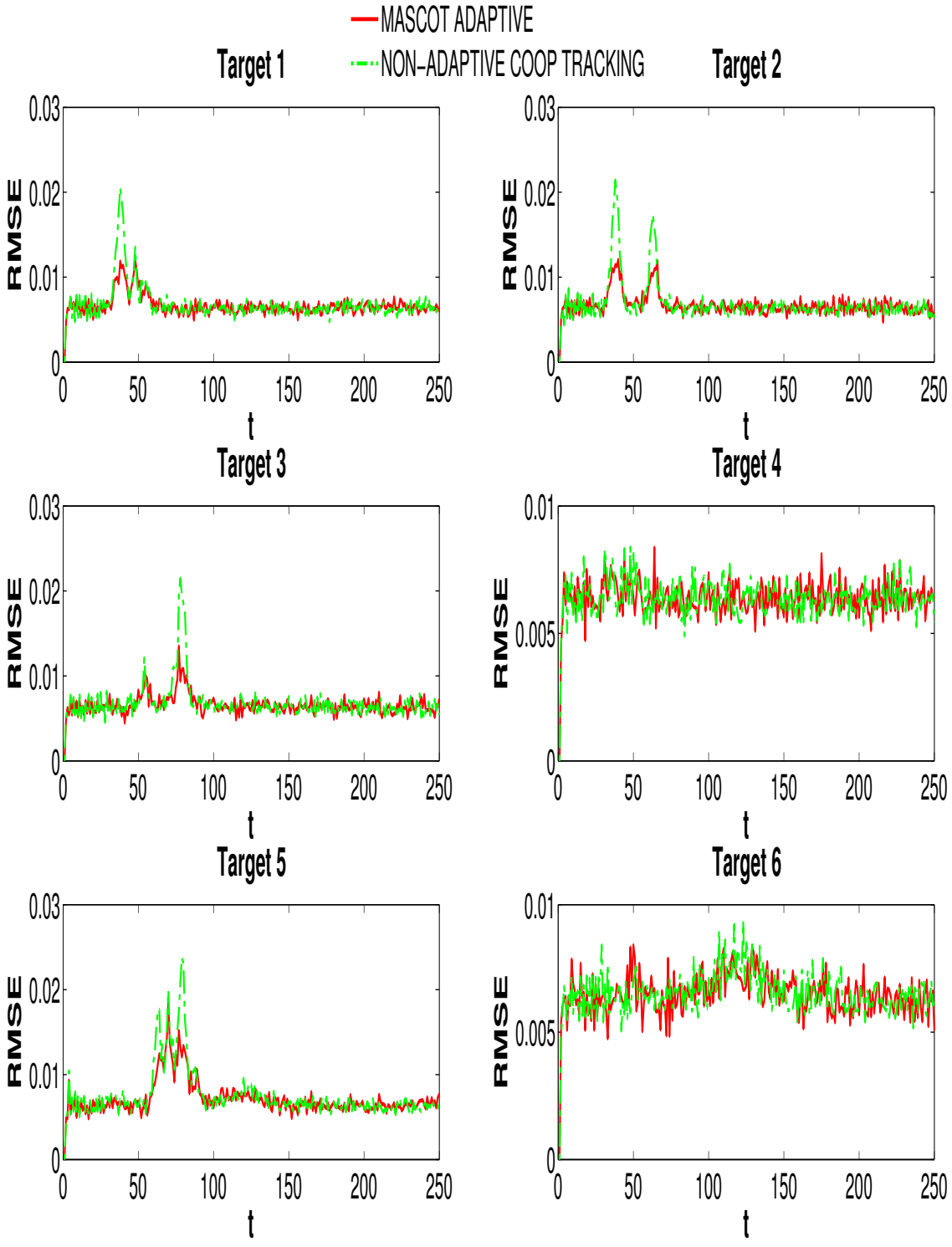


**Figure 2.20:** RMSE of Six-Target Full MASCOT Scenario with Comparison to Non-Adaptive Cooperative Filter.

# 3

# Target Tracking in the Presence of Interference

## 3.1 Overview

This chapter details research that has been conducted relating to a topic that, while similar in nature to the target application of the previous chapter, currently lies outside of the scope of the proposed MASCOT algorithm. Specifically, the focus is on maintaining successful tracking of a target state using a set of measurements which may be affected by unknown sources of interference. It is difficult to say too much here without referring to a specific model, so the following rather general form for a given scalar measurement $y_t$ taken at time $t$ is initially adopted here:

$$y_t = h(\mathbf{x}_t) + v_t + b_t \tag{3.1}$$

where $h(\mathbf{x}_t)$ is some function of the target state $\mathbf{x}_t$ (that is assumed to be moving in a 2D plane) at time $t$, $v_t$ is random sensor noise of which statistics (the pdf) are typically known, and $b_t$ is an unknown (random) interference component. With zero knowledge concerning $b_t$, one might argue that the problem is ill-posed and that $\mathbf{x}_t$ is not "estimable" based solely on $y_t$; indeed one cannot make rational inference concerning some unknown quantity based on an observation that is essentially "infinitely" variable. Thus we must proceed by assigning some model to the interference; obviously our resulting inference about $\mathbf{x}_t$ in reality will only be reasonable if the true behavior of $b_t$ is faithful to said model. Ideally, this model should be as general as possible, allowing us to cover a broad range of real interference sources, but again this must be balanced with the underlying intent to estimate $\mathbf{x}_t$; a model that is too general may not allow this to be done.

An obvious, and fairly rational decision to make in model selection for $b_t$ is to assume that this

interference is generated by a combination of *localized* interference sources, or:

$$b_t = \sum_{l=1}^{L,t} h\left(\mathbf{x}_{l,t}\right) \tag{3.2}$$

where $\mathbf{x}_{l,t}$ represents the location of the $l$-th interferer, and there are a total of $L_t$ (unknown) sources at time $t$. This represents an important, albeit challenging problem; the task of tracking a single target using measurements that are affected by other "foreign" interfering targets within the environment. This problem was addressed and solved successfully within the MASCOT framework, however there it is assumed that the number of interferers is always known and does not change with time. Here the problem is addressed from a more general viewpoint; assuming as little as possible about the interference. One approach towards a solution lies in attempting to simultaneously track the target of interest and perform detection/localization of these "interferers". This idea forms the basis of the approach in [1] within an algorithm which employs a PF-based approach and its details are now reviewed.

## 3.2 Interference Compensation through Source Localization

At time $t-1$, a representation of the posterior distribution of $\mathbf{x}_{t-1}$ is formed using the random measure $\chi_t = \{w_{t-1}^{(m)}, \mathbf{x}_{t-1}^{(m)}\}_{m=1}^{M}$. Suppose a prediction of the target state at time $t$ denoted $\tilde{\mathbf{x}}_t$ is then made based on the particle set. Assuming that no interference was present at $t-1$ and that the prediction $\tilde{\mathbf{x}}_t$ is reasonably accurate, a decision about the existence of interference in the new measurement $y_t$ can be made by forming the following detection criterion:

$$T\left(\tilde{\mathbf{x}}_t, y_t\right) \triangleq \left(y_t - h\left(\tilde{\mathbf{x}}_t\right)\right)^2 \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \gamma \tag{3.3}$$

where $\mathcal{H}_0, \mathcal{H}_1$ represent respectively the null and affirmative hypothesis concerning the existence of interference in $y_t$ and $\gamma$ is the related detection threshold. In theory, determination of $\gamma$ would proceed according to some desired performance criteria. For example [see pg. 65 in [45] ], maximizing the probability of detection ($P_D$) for a given false-alarm probability ($P_{FA} = \rho$) would produce the following constraint on $\gamma$:

$$\int_{\{(y_t, \tilde{\mathbf{x}}_t) \geq \gamma\}} f\left(y_t, \tilde{\mathbf{x}}_t; \mathcal{H}_0\right) dy_t d\tilde{\mathbf{x}}_t = \rho \tag{3.4}$$

Unfortunately, this represents a difficult, and likely intractable constraint to compute in practice. Theoretical determination of a proper threshold was not determined in the paper [1] and instead, a reasonable value for $\gamma$ based on a combination of the known measurement statistics and empirical observations was chosen in order to proceed with algorithm development.

In [1], the specific case of a single, static interference source with location denoted as $\mathbf{l}_t$ at time $t$

*At time $t^\star$ complete the steps*:

*S1*: Form an initial estimate of the interferer location $\tilde{\mathbf{l}}_{t^\star}$ based on the value of $\mathbf{y}_{t^\star} - \mathbf{h}\left(\tilde{\mathbf{x}}_{t^\star}\right)$ using a triangulation based approach.

*S2*: Propagate particles $\mathbf{x}_{t^\star}^{(m)} \sim f\left(\mathbf{x}_{t^\star}|\mathbf{x}_{t^\star-1}^{(m)}\right)$.

*S3*: For each particle $\mathbf{x}_{t^\star}^{(m)}$, generate an *interferer particle* $\mathbf{l}_{t^\star}^{(m)} \sim \mathcal{N}\left(\tilde{\mathbf{l}}_{t^\star}, \Sigma_{t^\star}\right)$ where $\Sigma_{t^\star}$ is a pre-specified *particle initialization* covariance matrix that expresses an uncertainty in the initial estimate $\hat{\mathbf{l}}_{t^\star}$.

*S4*: Compute particle weights according to $w_{t^\star}^{(m)} \propto w_{t^\star-1}^{(m)} f\left(\mathbf{y}_{t^\star}|\mathbf{x}_{t^\star}^{(m)}, \hat{\mathbf{l}}_{t^\star}^{(m)}\right)$.

*S5*: Normalize the weights and compute estimates:

$$\hat{\mathbf{x}}_{t^\star} = \sum_{m=1}^{M} w_{t^\star}^{(m)} \mathbf{x}_{t^\star}^{(m)} \quad \hat{\mathbf{l}}_{t^\star} = \sum_{m=1}^{M} w_{t^\star}^{(m)} \mathbf{l}_{t^\star}^{(m)} \quad \hat{\mathbf{\Sigma}}_{t^\star} = \sum_{m=1}^{M} w_{t^\star}^{(m)} \left(\mathbf{l}_{t^\star}^{(m)} - \hat{\mathbf{l}}_{t^\star}\right)\left(\mathbf{l}_{t^\star}^{(m)} - \hat{\mathbf{l}}_{t^\star}\right)^\top$$

*For $t \geq t^\star$*:

*S6*: Propagate particles $\mathbf{x}_t^{(m)} \sim f\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}\right)$ and $\mathbf{l}_t^{(m)} \sim \mathcal{N}\left(\hat{\mathbf{l}}_{t-1}, \Sigma_{t-1}\right)$.

*S7*: Repeat steps *S4* and *S5* replacing $t^\star$ with $t$.

*S8*: Resample if necessary.

**Table 3.1:** PF Algorithm for tracking a single target with static interference compensation

was considered. An RSS-type measurement model in the same form as equation (3.1) was adopted. The paper considered an environment with multiple sensors employed at separate locations each producing synchronized measurements that are collected in the vector $\mathbf{y}_t$; with a sufficient number of sensors, crude localization of a target can be performed based on a triangulation-style approach. Assuming a decision has been made that interference exists within $\mathbf{y}_{t^\star}$ (which is now generated through the vector function $\mathbf{h}(\mathbf{x}_{t^\star})$, i.e. $[\mathbf{y}_{t^\star}]_k = [\mathbf{h}(\mathbf{x}_{t^\star})]_k + [\mathbf{b}_t^\star]_k + [\mathbf{v}_t^\star]_k$ ), where $t^\star$ is the initial detection time of interference, then the algorithm can proceed as follows: Note that here the variable $\hat{\mathbf{l}}_t$ refers to an estimate of the interference location that was computed at time $t$, not its value at time $t$ (since the location is assumed to be static for all time). This algorithm can be viewed as a simplified application of the approach detailed in [12] for handling unknown constant parameters within a particle filter. A simulation study was conducted to investigate its performance under various conditions where reasonable accuracy was observed; see the original paper for detailed results. Though it was not specifically addressed in the paper, it should be fairly obvious that this algorithm can be extended in a relatively straightforward manner to deal with multiple static interferers, as long as it is assumed that only a single source can appear/disappear at any given time and that existing interferers have been sufficiently localized. It may seem at first glance that this would require an exponentially increasing number of particles as the total number of detected interferers increases similar to the dimensionality problem experienced in MTT, however this is not necessarily the case. As long as the last detected source (say $\mathbf{l}_{k,t}$) can be localized to sufficient

accuracy before another one ($\mathbf{l}_{k+1,t}$) is detected, then the particle set for $\mathbf{l}_{k,t}$ can be collapsed into a single point estimate $\hat{\mathbf{l}}_{k,t}$ or a Gaussian density as done in MASCOT, at the time of a new source detection, allowing a constant size particle set to be maintained.

## 3.3 Compensation by Dynamic Bias Tracking

While remaining a viable approach for the specific scenario discussed, the previous algorithm is limited in that it cannot handle dynamic interference sources. Incorporating interferer dynamics into the algorithm and attempting to proceed with source detection/localization as before simply yields an MTT tracking scenario, along with its associated curse of dimensionality. Another possible approach rests on the assumption of statistical independence of the target along with all existing interference sources (which is reasonable in most cases considered). Writing the measurement update equation that is typical in Bayesian tracking algorithms:

$$f(\mathbf{x}_t \mid \mathbf{y}_{1:t}) \propto f\left(\mathbf{y}_t \mid \mathbf{x}_t\right) f\left(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}\right) \tag{3.5}$$

Then the interference sources can be handled by marginalization, i.e. by *computing* the likelihood $(\mathbf{y}_t \mid \mathbf{x}_t)$ through:

$$(\mathbf{y}_t \mid \mathbf{x}_t) = \int f(\mathbf{y}_t \mid \mathbf{x}_t, \mathbf{l}_{1:L_t,t}) d\mathbf{l}_{1:L_t,t} \tag{3.6}$$

Note how this approach is similar in some ways to the discussion surrounding (2.37). In general, this represents a difficult, likely intractable, problem to solve. Nonetheless, this approach was pursued and seemingly effective approximations were developed to accomplish this in [46] for the case of specific passive sonar detection and tracking in the presence of interference.

Since it is assumed there is no specific interest in tracking the actual locations of all possible interferers, it is highly desirable to seek methods for coping with the corrupted measurements which do not require this to be done. Any algorithm which accomplishes this can reap dramatic savings in computational expense since it will not suffer from the dimensionality increase necessary in tracking a state space augmented with the interferer locations. The idea here is to express the measurement model for $\mathbf{y}_t$ in its original form as in (3.1) and treat $\boldsymbol{b}_t$ as a *dynamic bias component* of the measurement, essentially ignoring the underlying interference sources which generated it. This shifts the problem into the domain of a fairly well studied area; that of compensating for biased sensor measurements. Some related work on this subject includes [47],[48], [49], and [50], albeit in contexts outside of the current scope for various reasons. An interesting approach was also outlined in [51] although the author was unable to reproduce similar results and is concerned about the validity of the method in a general context.

A series of papers were written,([52],[53],[54]), that address sensor-bias within a particle-filtering

*Complete the following steps for time $t$:*

*S1*: Propagate $\mathbf{x}_t^{(m)} \sim f\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(m)}\right)$ and $\sigma_t^{(m)} \sim f\left(\sigma_t \mid \sigma_{t-1}^{(m)}\right)$

*S2*: Form $\mathbf{C}_{b_{t|t-1}}^{(m)} = \mathbf{C}_{b_{t-1|t-1}}^{(m)} + \left(\sigma_t^{(m)}\right)^2 \mathbf{I}$ and $\mathbf{S}_t^{(m)} = \mathbf{C}_{b_{t|t-1}}^{(m)} + \sigma_v^2 \mathbf{I}$

*S3*: Compute the bias estimates:

$$\hat{\mathbf{b}}_t^{(m)} = \hat{\mathbf{b}}_{t-1}^{(m)} + \left[\mathbf{C}_{b_{t|t-1}}^{(m)} \left(\mathbf{S}_t^{(m)}\right)^{-1}\right] \left(\mathbf{y}_t - \mathbf{h}\left(\mathbf{x}_t^{(m)}\right) - \hat{\mathbf{b}}_{t-1}^{(m)}\right)$$

*S4*: Form $\mathbf{C}_{b_{t|t}}^{(m)} = \left(\mathbf{I} - \left[\mathbf{C}_{b_{t|t-1}}^{(m)} \left(\mathbf{S}_t^{(m)}\right)^{-1}\right]\right) \mathbf{C}_{b_{t|t-1}}^{(m)}$

*S5*: Compute the weights $w_t^{(m)} \propto w_{t-1}^{(m)} f\left(\mathbf{y} \mid \mathbf{x}_t^{(m)}, \sigma_t^{(m)}\right)$ where,

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{h}\left(\mathbf{x}_t^{(m)}\right) + \hat{\mathbf{b}}_{t-1}^{(m)}, \mathbf{S}_t^{(m)}\right)$$

*S6*: Normalize the weights.

*S7*: Form the estimates $\hat{\mathbf{x}}_t = \sum_{m=1}^{M} w_t^{(m)} \mathbf{x}_t^{(m)}$ and $\hat{\sigma}_t = \sum_{m=1}^{M} w_t^{(m)} \sigma_t^{(m)}$.

**Table 3.2:** PF Algorithm for tracking a single target with unknown interference modeled as a dynamically varying bias.

framework using the concept of Rao-Blackwellization (see [55] and [56]). The work completed in these papers formed the basis for [57] which models the sensor-bias component as a random variable that obeys the dynamics:

$$\mathbf{b}_t = \mathbf{b}_{t-1} + \mid \sigma_t \mid \mathbf{g}_t \tag{3.7}$$

where $g_{n,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and

$$\sigma_t = \sigma_{t-1} + \epsilon_t \tag{3.8}$$

Here $\epsilon_t \sim \mathcal{N}(0, \sigma_e)$ is a perturbation to the parameter $\sigma_t$ whose absolute-value represents the standard deviation of the bias at time $t$.

Similar to the algorithm outlined for the case of static interference, a random measure $\{w_t^{(m)}, \mathbf{x}_t^{(m)}, \sigma_t^{(m)}\}_{m=1}^{M}$ is used to represent the target state posterior at time $t$. Then conditioned on the value of a specific particle, we have:

$$\mathbf{y}_t = \mathbf{h}\left(\mathbf{x}_t^{(m)}\right) + \mathbf{b}_t + \mathbf{v}_t \tag{3.9}$$

The measurement vector $\mathbf{y}_t$ is thus conditionally linear-Gaussian and the bias component vector can be estimated through Kalman-filtering. Assuming the measurement noise is $\mathbf{v}_t \sim \mathcal{N}\left(\mathbf{0}, \sigma_v^2 \mathbf{I}\right)$, and initial statistics for the bias as $[\mathbf{b}_0]_k \sim \mathcal{N}\left([\mathbf{b}_0]_k, \mathbf{C}_{b_0}\right)$ then the algorithm listed in table 3.2 was proposed.

## 3.4 Performance Results

The proposed algorithm was tested in a scenario similar to the one used for testing in the single-interference case as described in [1]. Essentially, a grid of sensors are on standby until detection of a target is made. Upon detection, four of the standby sensors with the largest signal power are assigned to track the target, and maintain position uniformly spaced about a circle with radius 3 m centered about a prediction of the target in the current time step. The specific dynamic model that was considered for the target is:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t$$

where $\mathbf{x}_t$ is the state vector defined as $\mathbf{x}_t = [x_{1,t}\ x_{2,t}\ \dot{x}_{1,t}\ \dot{x}_{2,t}]^\top$, with the first two components representing position coordinates and the second two representing the velocity of the target. The component $\mathbf{u}_t$ is a $2 \times 1$ vector representing the target process noise and is assumed $\sim \mathcal{N}\left(0, \sigma_u^2\right)$. The matrices $\mathbf{A}$ and $\mathbf{B}$ were defined as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_2 & T_s\mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \frac{T_s^2}{2}\mathbf{I}_2 \\ T_s\mathbf{I}_2 \end{bmatrix}$$

with $T_s$ as the sampling period. The measurement component produced by a target, $\mathbf{h}\left(\mathbf{x}_t\right)$, is modeled as:

$$\left[\mathbf{h}\left(\mathbf{x}_t\right)\right]_k = \frac{\Phi d_0^\alpha}{\|\mathbf{s}_{k,t} - \mathbf{x}_t^{1:2}\|^\alpha} \tag{3.10}$$

where $\mathbf{s}_{k,t}$ is the $k$-th sensor location, $\Phi$ is the emitted signal power by the target measured at a distance $d_0$, and $\alpha$ is a path-loss coefficient depending on the physical transmission medium and assumed known. Parameters during simulation were set to $M = 700$, $\sigma = 10^{-4}$, and $\sigma_v = 10^{-3}$, $\Phi = 1$, $d_0 = 1$, and $\alpha = 2$.

Simulations were first run using "synthetic" bias components which obeyed the model in (3.7). One specific scenario examined involves "turn-on" of the interference components approximately 50 time steps after tracking of the target has stabilized. Figure 3.1 shows a single trajectory plot and 3.2 shows the associated location error (norm of the location coordinates) of a single realization with the perturbation parameter set to $\sigma_e^2 = 0.02$ for the "interference-compensated" algorithm described, alongside a "blind" particle filter that does not account for the interference. It is clear for this realization that the uncompensated filter immediately fails upon turn-on, while reasonable tracking performance is maintained for the compensated filter. The actual interference bias for the first sensor along with the corresponding estimates for this trajectory obtained by the proposed method are also shown in figure 3.3; note how there is a relatively large variance in this signal which may not be entirely realistic for "real" localized interference sources.

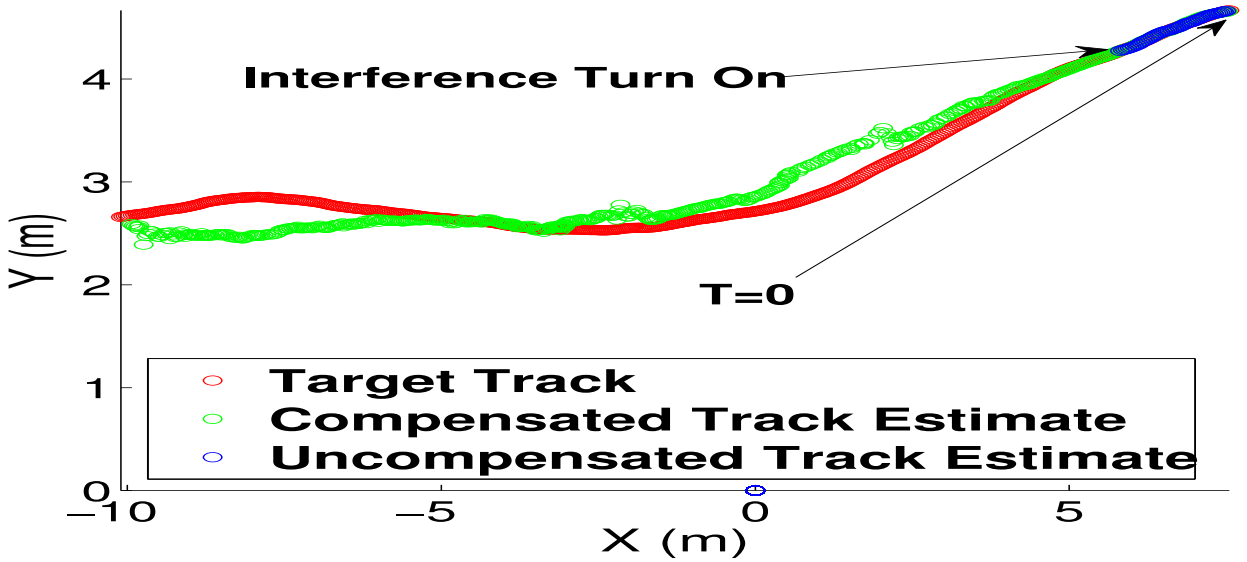


**Figure 3.1:** Single Realization Trajectory Plot for the Interference Compensated Tracker

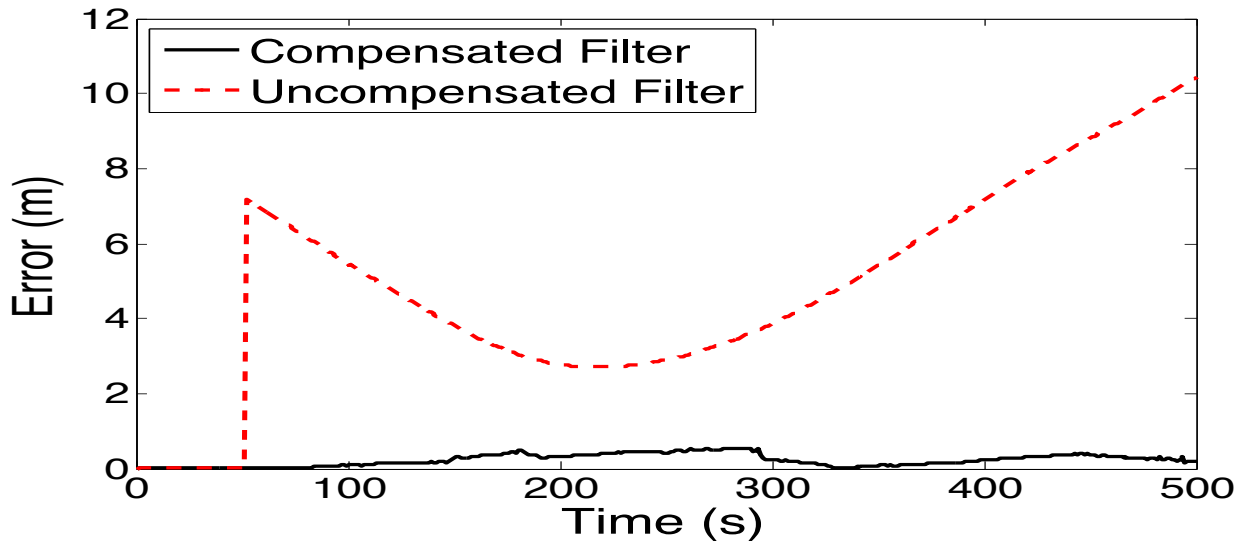


**Figure 3.2:** Norm Error Plot for a Single Realization of the Interference Compensated Tracker

A simulation to compute the RMSE (mean norm error of the target location) over 100 trials was then conducted for various values of $\sigma_e$ and is plotted in 3.4. It can be seen that there is a gradual increase in error over time, particularly with large values of $\sigma_e$; this is not entirely unexpected since there is a higher likelihood that the instantaneous variance of the bias ($\sigma_t^2$) component will be higher at later times. This does represent a possible drawback to the model, which allows for an unbounded increase in $\sigma_t$, and is not realistic for an environment with a finite number of interference sources.

Performance of the proposed filter was then tested against a "real" scenario involving 10 sources

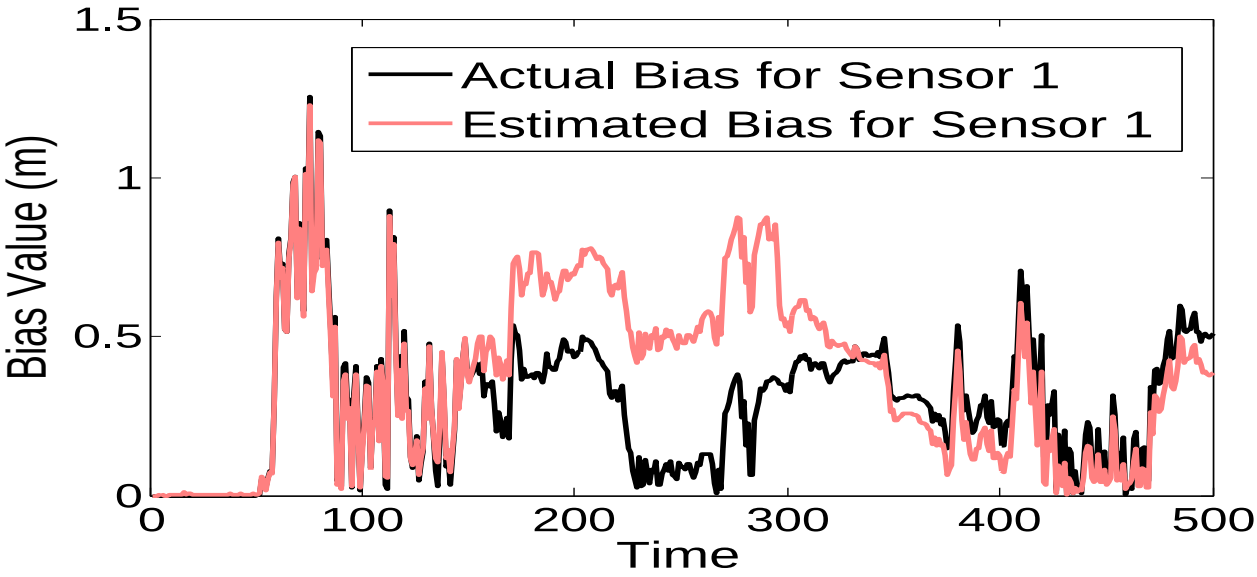


**Figure 3.3:** Measurement Bias Component being tracked for the first sensor

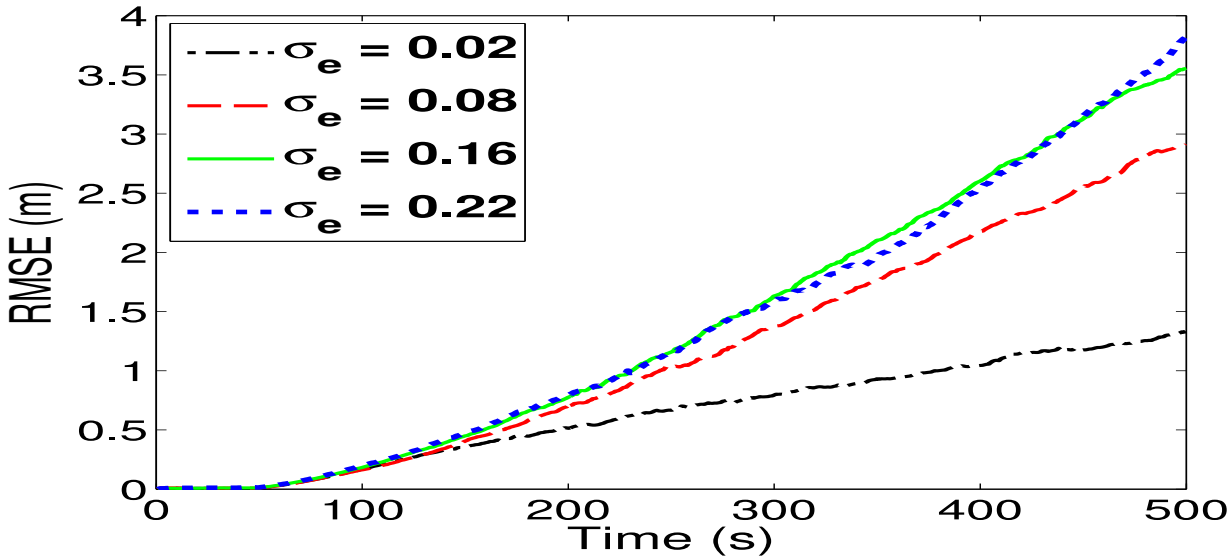


**Figure 3.4:** RMSE over 100 trials for a synthetic bias with various $\sigma_e$

of interference that simultaneously deploy from the target location at a particular time and move freely according to the same model used for the target. A value of $\sigma_e = 2$ was empirically determined to perform best for this particular situation. Figure 3.5 (where the interferer's are marked with 'x's) shows a single trajectory realization, where it can be seen that while the non-compensated filter fails immediately upon interference deployment, the compensated algorithm maintains reasonable tracking for quite some time.

Simulation of the RMSE over 100 trials for this case was also conducted, where it was found that a gradual accumulation of error over time occurs similarly to when a synthetic bias is used;

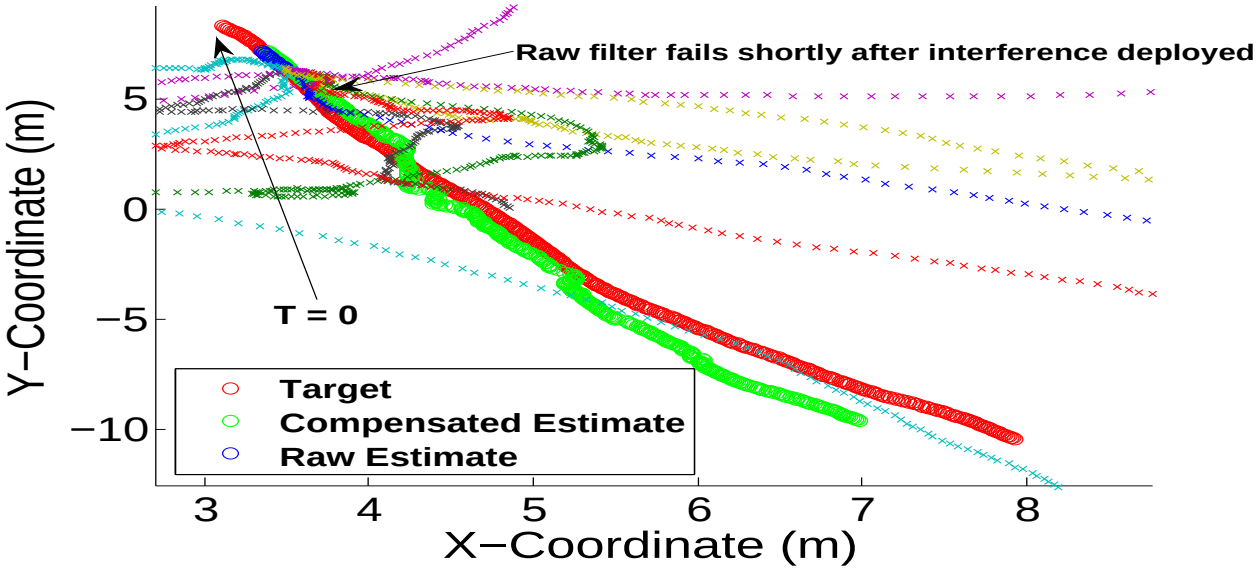


**Figure 3.5:** 10 Real Interference Sources Deployed From the Target Location

this is plotted in figure 3.6.

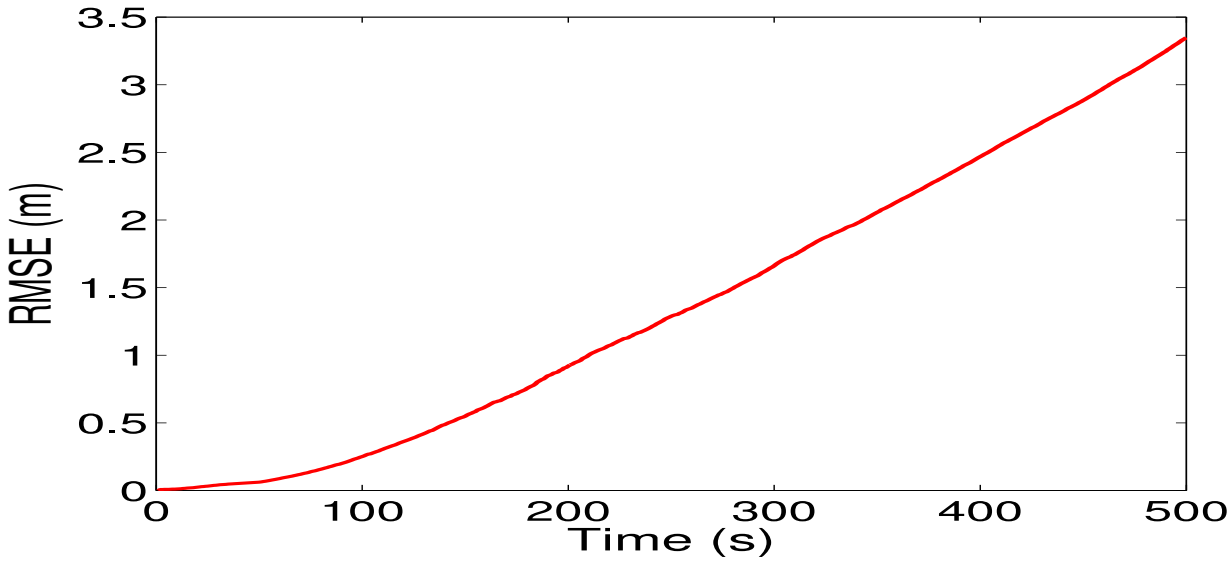


**Figure 3.6:** RMSE Over 100 Trials for the 10-Interferer Scenario

## 3.5 Concluding Remarks

Based on the preceding results, particularly in the case of a large number of real interference sources, this algorithm does appear to have potential as a viable solution to the problem. There are however, still some significant issues that would need to be resolved. Namely, the algorithm

exhibits a high sensitivity to process/measurement noise and quickly degrades at higher levels. Furthermore, it is not clear what the optimal value for the perturbation parameter $\sigma_e$ should be, although one could perform marginalization of this parameter to eliminate the need for a precise value. Assuming these issues can be adequately resolved, an immediate application would be within the MASCOT framework discussed in Section 2 as a means for *initialization* of individual agents. To clarify this concept further, recall that MASCOT relies on cooperation between agents that are assumed to possess fairly accurate estimates of their own state partitions. At times, particularly upon initialization and shortly thereafter, this may not be true in general and can severely limit the best attainable performance with MASCOT, since agents must be able to track their own targets in the presence of a potentially large number of unknown interfering targets. Employment of this algorithm would enable a coarse tracking of the targets to be maintained until the agents are able to negotiate properly and form more stable/accurate estimates.

# 4

# Multi-Mobile-Sensor Target Tracking with Asynchronous Measurements

## 4.1 Overview

A key underlying assumption up to this point involves the *perfect synchronization* of measurements originating from different sensors. It should be fairly obvious that performance degradation can occur if a violation of this assumption is not accounted for; fusion of measurements that were taken at different instants, as if they occurred simultaneously, can result in considerable error if the target is not in the same state at each point. Since there is frequently some level of unavoidable asynchronism present in a real system, attributed to a number of sources including communication delays across a transmission medium, sensor processing delays, and timing clock drift, this is an important issue to consider in the design of any practical tracking system. When dealing with a multi-sensor environment, there are two distinct types of asynchronicity that need to be addressed. However, in order to properly express this distinction, it first needs to be mentioned what exactly is meant by a synchronous set of measurements. Suppose that a group of $K$ sensors each take measurements at periodic intervals of duration $T$, with each sensor immediately transmitting its own measurement back to a centralized data fusion center. A fully synchronous system involves the combination of alignment of each sensor's measurement interval (all sensors take measurements at $t = kT$ for integer $k$), along with *reception of these measurements at the fusion center within the same measurement interval.* Out-of-sequence measurements (OOSM) are the first type of asynchronicity which arises when the latter condition is not satisfied; while sensor measurements were taken synchronously, some may arrive at the fusion center at a later interval, for example, a measurement from one of the sensors taken at $t = kT$ is not received until $t = (k + 2)T$. The question arises in this case as to how the late-arriving information should be incorporated (if at all) when the estimated target track may have already been updated using measurements from other

sensors that were received on-time; i.e. it is doubtful that there would be a significant benefit from updating a track using highly delayed information. The second type of asynchronicity arises when the sensor measurement intervals themselves are not aligned, i.e. sensor 1 takes measurements at $t = kT$ while sensor 2 takes them at $t = kT + \tau$. While this distinction between the two types is noted here, they can obviously occur together and further complicate resolution of this issue.

This has become an increasingly active area of study and significant progress has been made in tackling the problem. In the case of linear-Gaussian tracking, the optimal solution for updating the current target state estimate and covariance matrix estimate based on a late-arriving measurement by "retrodiction" of the current estimate to the time at which the measurement was taken, was derived in [58]. In [59] a number of algorithms were presented that focus on decentralized tracking, also dealing with linear-Gaussian tracking, which address "multi-lag" (measurements received that correspond to instants that occurred multiple intervals before the current one) and extend the solution to the multiple-model tracking case (for example, tracking a target which may at any time conduct one of several predefined maneuvers). Some interesting results were presented in [60] which show how sensor measurement instances can be intentionally staggered to benefit estimation performance; a specific algorithm was proposed again in the context of linear-Gaussian tracking. In [61] an algorithm is proposed which selectively fuses OOSM, choosing not to update track estimates based on those measurements which can potentially harm rather than benefit performance. A treatment of asynchronous measurements within the context of particle filtering (which can perform reasonably in environments where the other algorithms will deteriorate due to the linear-Gaussian assumptions made) is relatively sparse in the literature; [62] and [63] are exceptions but utilize techniques that are rather heuristic in nature. Most relevant is the approach outlined in [64] which introduces a reformulation of the original tracking problem that is more suitable for asynchronous measurements. The algorithm presented is in the context of a static sensor network with binary measurements; this forms the basis for the work in [3] which will now be presented.

## 4.2 Proposed Solutions

Let us consider an environment with a single target, and in which multiple agents exist tasked with tracking this target, each operating asynchronously and independently from one another. For each agent, there is an associated *base time*; which represents the instants at which measurements are taken by an agent. The constraint in this scenario is imposed that each agent estimates the target state at its own base time at a minimum (a particular agent may also form estimates at other agent's base times). A new notation is here introduced which defines each agent's base time in terms of "absolute time"; within the absolute interval of time $t = (k-1)T$ to $t = kT$, the base time (or corresponding measurement instant) of agent $i$ is $t = (k-1)T + \tau_i$. For simplicity, agents

are notationally ordered such that $\tau_{i-1} \leq \tau_i$ for all $i$. Variables which take on a value at this time instant will have a subscript $(t, i)$ referring to the corresponding base time for the $i$-th agent.

In terms of absolute time, the target dynamics are modeled according to:

$$\mathbf{x}_{t_2} = \mathbf{A}\left(t_2, t_1\right) \mathbf{x}_{t_1} + \mathbf{u}_{t_2,t_1} \tag{4.1}$$

where $t_2 \geq t_1$ represent two instants of absolute time, $\mathbf{x}_{t_2}$ is a $4 \times 1$ vector defined by $\mathbf{x}_{t_2} = [x_{1,t_2},\, x_{2,t_2},\, \dot{x}_{1,t_2},\, \dot{x}_{2,t_2}]^\top$ with $x_{1,t_2}$ and $x_{2,t_2}$ the location coordinates of the target at time $t = t_2$ and $\dot{x}_{1,t_2}, \dot{x}_{2,t_2}$ the corresponding velocity coordinates. The symbol $\mathbf{A}(t_2, t_1)$ denotes a known $4 \times 4$ transition matrix, defined by

$$\mathbf{A}(t_2, t_1) = \begin{bmatrix} \mathbf{I}_2 & (t_2 - t_1)\mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix} \tag{4.2}$$

with $\mathbf{I}_2$ and $\mathbf{0}_2$ as the identity and zero $2 \times 2$ matrices, respectively. The state noise $\mathbf{u}_{t_2,t_1}$ is a $4 \times 1$ vector whose distribution is assumed Gaussian with covariance matrix

$$\mathbf{Q}(t_2, t_1) = \sigma_u^2 \begin{bmatrix} \frac{(t_2-t_1)^3}{3} & \frac{(t_2-t_1)^2}{2} & 0 & 0 \\ \frac{(t_2-t_1)^2}{2} & (t_2 - t_1) & 0 & 0 \\ 0 & 0 & \frac{(t_2-t_1)^3}{3} & \frac{(t_2-t_1)^2}{2} \\ 0 & 0 & \frac{(t_2-t_1)^2}{2} & (t_2 - t_1) \end{bmatrix} \tag{4.3}$$

where $\sigma_u^2$ is the equivalent continuous time process noise intensity. Now using the notation defined before, and assuming a total of $N$ agents exist, the target dynamics with respect to agent base times can be expressed as:

$$\begin{aligned} \mathbf{x}_{t,1} =& \mathbf{A}\left(t + \tau_1, t - T + \tau_N\right) \mathbf{x}_{t-1,N} + \mathbf{u}_{t+\tau_1,t-1+\tau_N} \\ \mathbf{x}_{t,i} =& \mathbf{A}\left(t + \tau_i, t + \tau_{i-1}\right) \mathbf{x}_{t,i-1} + \mathbf{u}_{t+\tau_i,t+\tau_i-1} \\ & i = 2, \cdots, N \\ & t = kT, \qquad \tau_1 < \tau_2 < \cdots < \tau_N < T \end{aligned} \tag{4.4}$$

Within each time interval, agent $i$ takes a measurement of the target at its own base time as described by,

$$y_{t,i} = \frac{\Phi d_0^\alpha}{\|\mathbf{s_{t,i}} - \mathbf{l}_{t,i}\|^\alpha} + \mathbf{v}_{t,i} \tag{4.5}$$

where $\mathbf{l}_{t,i} = [x_{1,t,i}\, x_{2,t,i}]^\top$ is the location of the target at time instant $(k-1)T + \tau_i$, $\mathbf{s}_{t,i}$ is the location of the $i$-th agent (sensor) at time instant $(k-1)T + \tau_i$, $\Phi$ is the emitted signal power of the target measured at distance $d_0$, and $\alpha$ is the path-loss coefficient which depends on the transmission medium and is assumed known. The observation noise $\mathbf{v}_{t,i}$ can have an arbitrary (known) distribution but is here assumed to be Gaussian with variance $\sigma_v^2$.

In the scenario considered here, agents cooperate by broadcasting their own measurements to the rest of the network. An agent then uses its own measurement along with others received (which were taken at different base times) to estimate the target location. There are four distinct possibilities of how measurements can be fused and these are illustrated in figure 4.1. Note that all figures used here have been taken directly from [3]; without loss of generality, it is assumed that there are a total of 4 agents in the environment. Notice in each case there are actually 4 unknowns that need to be represented; the target state $\mathbf{x}_{t,i}$ at each of the four possible agent base times.

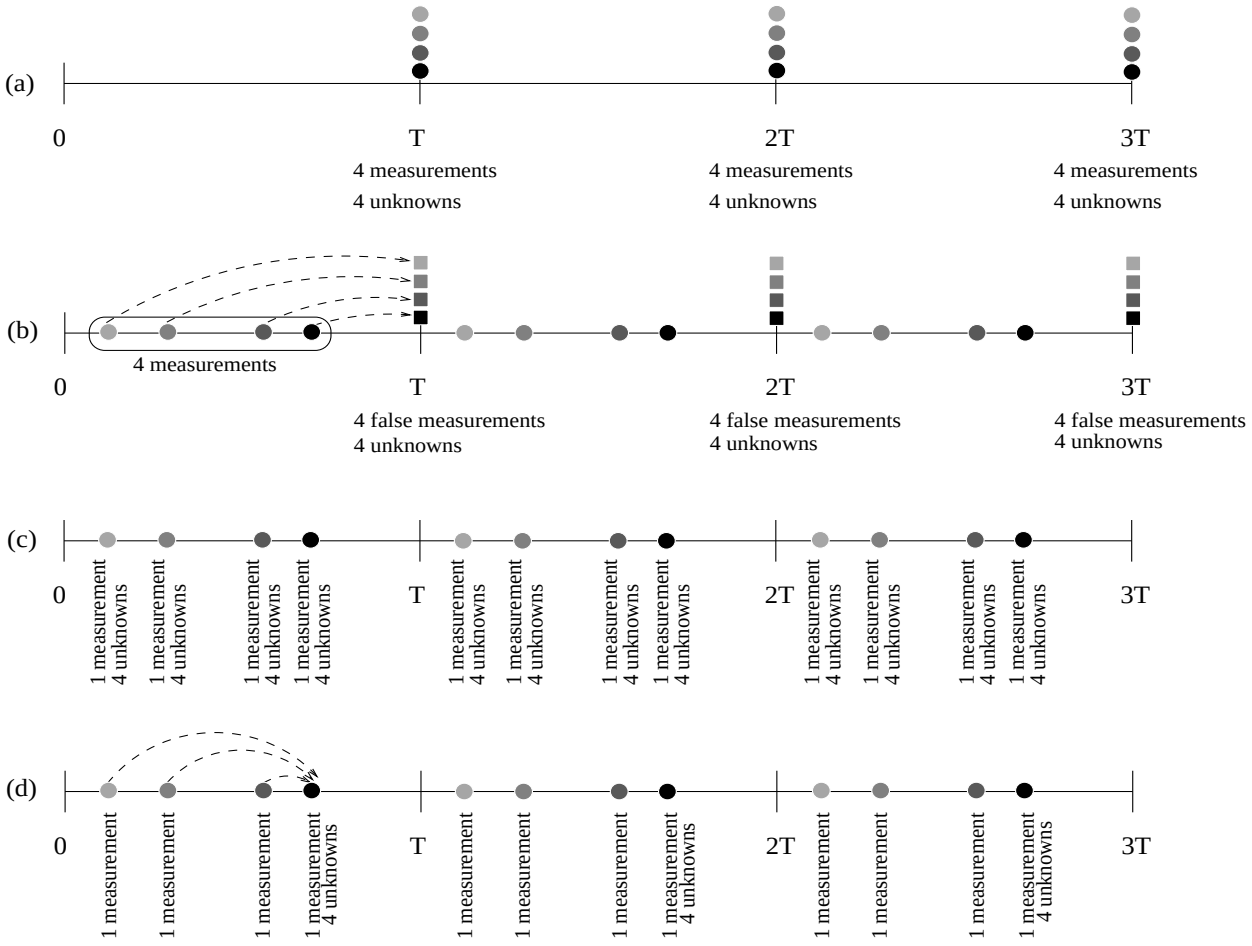


**Figure 4.1:** (a) "True" asynchronous method (b) "False" asynchronous method (c) asynchronous sequential method (d) asynchronous batch method

- *True Synchronous Method:* This corresponds to an alignment of agent's base times, i.e. $\tau_i = \tau_j$ for all $i$ and $j$. A basic particle filter is used to track the target for each agent in this case. Specifically, agent $i$ forms the random measure $\{\mathbf{x}_{t,i}^{(m)}, w_{t,i}^{(m)}\}_{m=1}^{M}$ representing the target posterior at the agent's base time. Propagation of the particles at each time instant and

computation of the weights proceeds as:

$$\mathbf{x}_{t,i}^{(m)} \sim f\left(\mathbf{x}_{t,i} \mid \mathbf{x}_{t-1,i}^{(m)}\right)$$
$$w_{t,i}^{(m)} \propto w_{t-1,i}^{(m)} \prod_{i=1}^{N} f\left(y_{t,i} \mid \mathbf{x}_{t,i}^{(m)}\right) \tag{4.6}$$

Since this represents fusion of synchronous measurements, this situation is used as a benchmark for the remaining scenarios.

- *False Synchronous Method:* Here the agent base times are misaligned, however each agent proceeds with estimation *as if all the measurements were at that agent's own base time*, i.e., in an identical manner as the True Synchronous Method. This scenario allows us to examine the resulting performance degradation from incorrectly assuming measurement synchronicity.

- *Asynchronous Sequential Method:* Agent base times are again misaligned, however in this case a particular agent updates its estimate of the target state each time it takes or receives a measurement. In other words, agent $i$ estimates the state $\mathbf{x}_{t,i}$ along with $\mathbf{x}_{t,j}$ for all $j \neq i$. Thus agent $i$ forms the random measure of the target posterior $\{\mathbf{x}_{t,j}^{(m)}, w_{t,j}^{(m)}\}_{m=1}^{M}$ where $t = kT$ for integer k and $j = 1, \cdots, N$. The particles at each agent are propagated according to:

$$\mathbf{x}_{t,1}^{(m)} \sim f\left(\mathbf{x}_{t,1} \mid \mathbf{x}_{t-1,N}^{(m)}\right)$$
$$\mathbf{x}_{t,j}^{(m)} \sim f\left(\mathbf{x}_{t,j} \mid \mathbf{x}_{t,j-1}^{(m)}\right), \quad j = 2, \cdots, N \tag{4.7}$$

and the particle weights of each agent at each base time instant $(k-1)\,T + \tau_j$ follow:

$$w_{t,1}^{(m)} \propto w_{t-1,N}^{(m)} f\left(y_{t,1} \mid \mathbf{x}_{t,1}^{(m)}\right)$$
$$w_{t,j}^{(m)} \propto w_{t,j-1}^{(m)} f\left(y_{t,j} \mid \mathbf{x}_{t,j}^{(m)}\right), \quad j = 2, \cdots, N \tag{4.8}$$

To further clarify these expressions, it is reiterated that agent $i$ updates its random measure at every base time instant. If the current time corresponds to the first base time or $t = (k-1)T + \tau_1$, then agent $i$ propagates its previous particle set which actually corresponds to the last base time in the previous absolute time interval, or $t = (k-2)T + \tau_N$. For other base times, agent $i$ propagates its previous particle set that now corresponds with the preceding base time within the current absolute time interval.

- *Asynchronous Batch Method:* Once again agent base times are misaligned, but in this situation a particular agent updates its target estimate only at its own base time (it estimates the state $\mathbf{x}_{t,i}$ only) by fusing its own current measurement with the most recent measurements it received from other agents. Particles for agent $i$ at its base time are propagated as:

$$\mathbf{x}_{t,i}^{(m)} \sim f\left(\mathbf{x}_{t,i} \mid \mathbf{x}_{t-1,i}^{(m)}\right), \quad m = 1, 2, \cdots, M \tag{4.9}$$

Since an agent does not estimate the target state at other base times aside from its own, it cannot directly fuse measurements together as was done in the asynchronous sequential method. Specifically, agent $i$ can no longer compute $f\left(y_{t,j} \mid \mathbf{x}_{t,j}^{(m)}\right)$ for $j \neq i$ as in (4.8). One way to approach this is to replace $\mathbf{x}_{t,j}^{(m)}$ in this factor with an estimate $\hat{\mathbf{x}}_{t,j}^{(m)}$ based on each of the agent's current particles, or

$$\hat{\mathbf{x}}_{t,j}^{(m)} = \mathbb{E}\left[\mathbf{x}_{t,j} \mid \mathbf{x}_{t,i}^{(m)}\right] \tag{4.10}$$

where $\hat{\mathbf{x}}_{t,j}^{(m)}$ can be viewed as a prediction of the target state at time $t = (k-1)T + \tau_j$, if $\tau_j > \tau_i$ or a *retrodiction* if $\tau_j < \tau_i$. This allows the agent's measurements to be properly fused via the modified weight update equation:

$$w_{t,i}^{(m)} \propto w_{t-1,i}^{(m)} f\left(y_{t,i} \mid \mathbf{x}_{t,i}^{(m)}\right) \prod_{n=1, n\neq i}^{N} f\left(y_{t,n} \mid \hat{\mathbf{x}}_{t,n}^{(m)}\right) \tag{4.11}$$

This solution is hereafter labeled as *Batch-1*.

Another alternative to this approach rests on the concept of an "asynchronous particle filter" that was developed in [64]. This approach consists of modifying which distribution the agent's particle filter is a representation of. Specifically, for agent $i$, the random measure $\{\mathbf{x}_{t,i}^{(m)}, w_{t,i}^{(m)}\}_{m=1}^{M}$ now represents the density $f\left(\mathbf{x}_{t,i}^{(m)} \mid \mathbf{y}_{1:t}\right)$, where $\mathbf{y}_t$ is a vector representing the collection of measurements that were made within the time interval from $(k-1)T$ to $kT$ and is expressed as $\mathbf{y}_t = [y_{t,1}, y_{t,2}, \cdots, y_{t,N}]^\top$. It is fairly easy to show that the weight update equation needs to be modified to:

$$w_{t,i}^{(m)} \propto w_{t-1,i}^{(m)} f\left(\mathbf{y}_t \mid \mathbf{x}_{t,i}^{(m)}, \mathbf{x}_{t-1,i}^{(m)}\right) \tag{4.12}$$

since the vector $\mathbf{y}_t$ depends not only on the state of the target at time $(k-1)T + \tau_i$, or $\mathbf{x}_{t,i}^{(m)}$, but also on the target's state at the base time within the previous interval, or $\mathbf{x}_{t-1,i}^{(m)}$ , since the vector $\mathbf{y}_t$ contains measurements that were taken at "intermediate" times between the times corresponding to these two states. For the sake of tractability a similar approximation as in [64] is used for this likelihood. Specifically, the joint likelihood is approximated as the factorization:

$$f\left(\mathbf{y}_t \mid \mathbf{x}_{t,i}^{(m)}, \mathbf{x}_{t-1,i}^{(m)}\right) \approx \prod_{n=1}^{N} f\left(y_{t,n} \mid \mathbf{x}_{t,i}^{(m)}, \mathbf{x}_{t-1,i}^{(m)}\right) \tag{4.13}$$

and each term is then decomposed according to

$$f\left(y_{t,n} \mid \mathbf{x}_{t,i}^{(m)}, \mathbf{x}_{t-1,i}^{(m)}\right) = \int f\left(y_{t,n} \mid \mathbf{x}_{t,n}\right) f\left(\mathbf{x}_{t,n} \mid \mathbf{x}_{t,i}^{(m)} \mathbf{x}_{t-1,i}^{(m)}\right) d\mathbf{x}_{t,n} \tag{4.14}$$

With the state process noise vector assumed to be Gaussian as mentioned earlier in the model definition, it was shown in [64] that the transition density $f\left(\mathbf{x}_{t,n} \mid \mathbf{x}_{t,i}^{(m)} \mathbf{x}_{t-1,i}^{(m)}\right)$, is

also a Gaussian distribution. However, evaluation of the integral in (4.14) for an arbitrarily distributed measurement noise remains intractable. The approach that was taken in [3] in an attempt to handle this was to argue that for small process noise intensity $\sigma_u^2$, the transition density is sharply peaked and will admit a fairly accurate approximation of (4.14) using Monte Carlo integration with relatively few sample points. So to approximate (4.14), an intermediate random measure for *each particle* $\mathbf{x}_{t,i}^{(m)}$ is formed as $\{\hat{\mathbf{x}}_{t,j}^{(k,m)}\}_{k=1}^{K}$ where

$$\hat{\mathbf{x}}_{t,j}^{(k,m)} \sim f\left(\mathbf{x}_{t,j} \mid \mathbf{x}_{t,i}^{(m)}, \mathbf{x}_{t-1,i}^{(m)}\right) \tag{4.15}$$

This set of intermediate particles acts as a representation of the "intermediate" transition density (evaluated at a given particle $\mathbf{x}_{t,i}^{(m)}$, specifically:

$$\left[f\left(\mathbf{x}_{t,j} \mid \mathbf{x}_{t,i}, \mathbf{x}_{t,i}\right)\right]_{\substack{\mathbf{x}_{t,i}=\mathbf{x}_{t,i}^{(m)} \\ \mathbf{x}_{t-1,i}=\mathbf{x}_{t-1,i}^{(m)}}} \approx \sum_{k=1}^{K} \delta\left(\mathbf{x}_{t,j} - \hat{\mathbf{x}}_{t,j}^{(k,m)}\right) \tag{4.16}$$

yielding an approximation to (4.14) for the $j$-th base time as:

$$\int f\left(y_{t,j} \mid \mathbf{x}_{t,j}\right) f\left(\mathbf{x}_{t,j} \mid \mathbf{x}_{t,i}^{(m)} \mathbf{x}_{t-1,i}^{(m)}\right) d\mathbf{x}_{t,j} \approx$$
$$\sum_{k=1}^{K} f\left(y_{t,j} \mid \mathbf{x}_{t,j}^{(k,m)}\right) f\left(\mathbf{x}_{t,j}^{(k,m)} \mid \mathbf{x}_{t,i}^{(m)} \mathbf{x}_{t-1,i}^{(m)}\right) \triangleq \hat{w}_{t,j}^{(k,m)} \tag{4.17}$$

This expression allows (4.12) to be computed as:

$$w_{t,i}^{(m)} \propto w_{t-1,i}^{(m)} f\left(y_{t,i} \mid \mathbf{x}_{t,i}^{(m)}\right) \prod_{j=1, j\neq i}^{N} \left(\sum_{k=1}^{K} \hat{w}_{t,j}^{(k,m)}\right) \tag{4.18}$$

With (4.18) the PF algorithm can then proceed as usual, i.e., particle propagation followed by weight update and possible resampling. This solution is hereafter labeled as *Batch-2*.

## 4.3 Performance Results

The algorithms discussed were tested in a similar setting as in Chapter 3 and initially outlined in [1], i.e., a single target is tracked using 4 agents which are deployed at each time on a ball of radius 3m centered at the predicted target location. The process noise intensity $\sigma_u^2$ was set to 0.005 and the measurement noise variance to 0.005 for all experiments. Other values of $\sigma_v^2$ were tested and it was found that as $\sigma_v^2$ increases, the asynchronism of the measurements becomes less relevant for the performance since the error due to the noise becomes larger than the error due to incorrectly assuming that the measurements occurred at the same instant.

To evaluate the performance deterioration that can occur as a result of asynchronicity, an initial

simulation experiment was conducted which runs only the "true" and "false" synchronous methods alongside one another (no compensation) for three specific scenarios each involving different sets of agent base times and all using 400 particles. Specifically, recalling that the $i$-th base time is $(k-1)T+\tau_i$, the set of $\tau_i$ for scenario 1 is $\{0, 0.25, 0.50, 0.75\}$; for scenario 2 is $\{0, 0.1, 0.5, 0.51\}$; and for scenario 3 is $\{0, 0.05, 0.15, 0.75\}$. An illustration of the specific time offsets used in each scenario is shown in figure 4.2 and a comparison in RMSE performance over 100 trials for each of the sets is illustrated in figure 4.3.

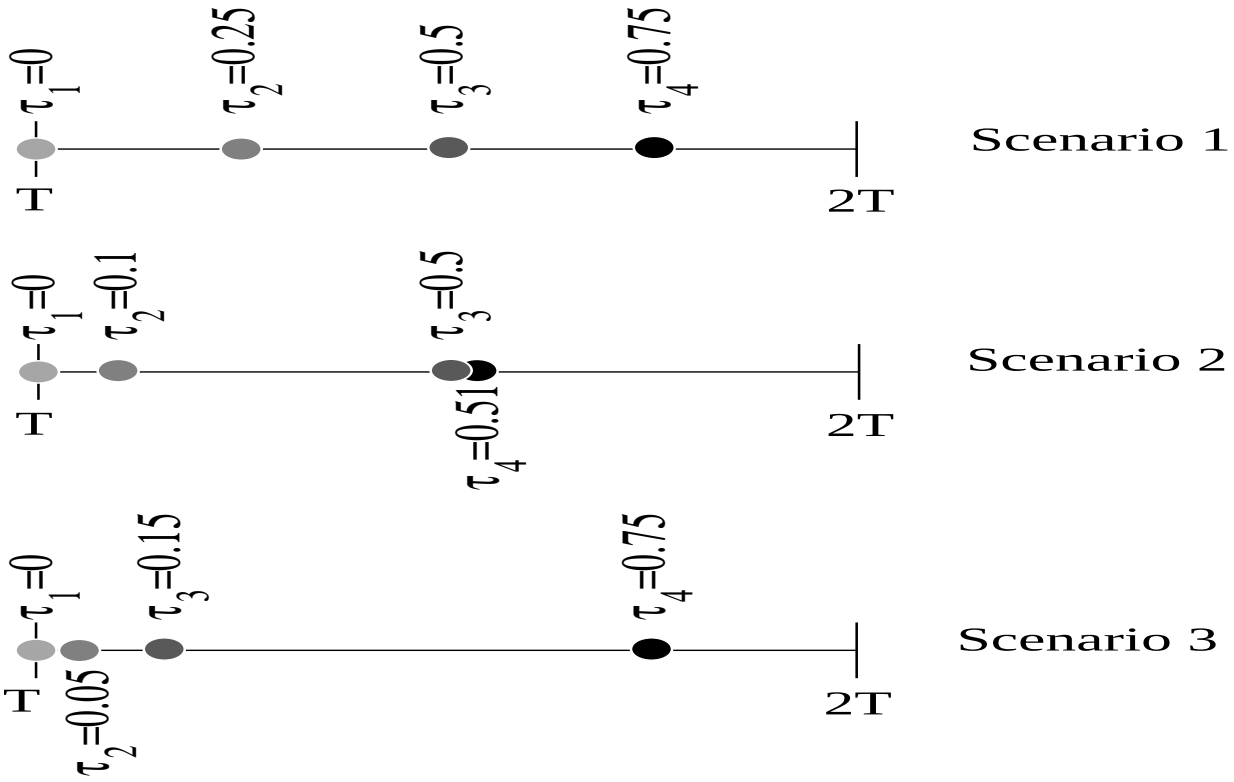


**Figure 4.2:** Time offset illustration for 3 asynchronous simulation scenarios

It can be seen that scenario 1, corresponding to evenly spaced measurements throughout the interval, yields the greatest performance deterioration for the "false" method (PF that incorrectly assumes measurements are synchronized). Although the performance is also degraded with the remaining two sets of measurement times, there is no significant difference between them. Figure 4.4 shows performance of the proposed algorithms (note 10 particles are used in the integration step for the Batch-2 method) applied to scenario 1; it is interesting to note that performance is roughly the same for each different method in this case.

To evaluate computational load of the various algorithms, the normalized run-time (which is defined as the ratio between the average run-time for the considered algorithm over the average run-time obtained using the "true" synchronous method as a reference, i.e., a value of 100% indicates

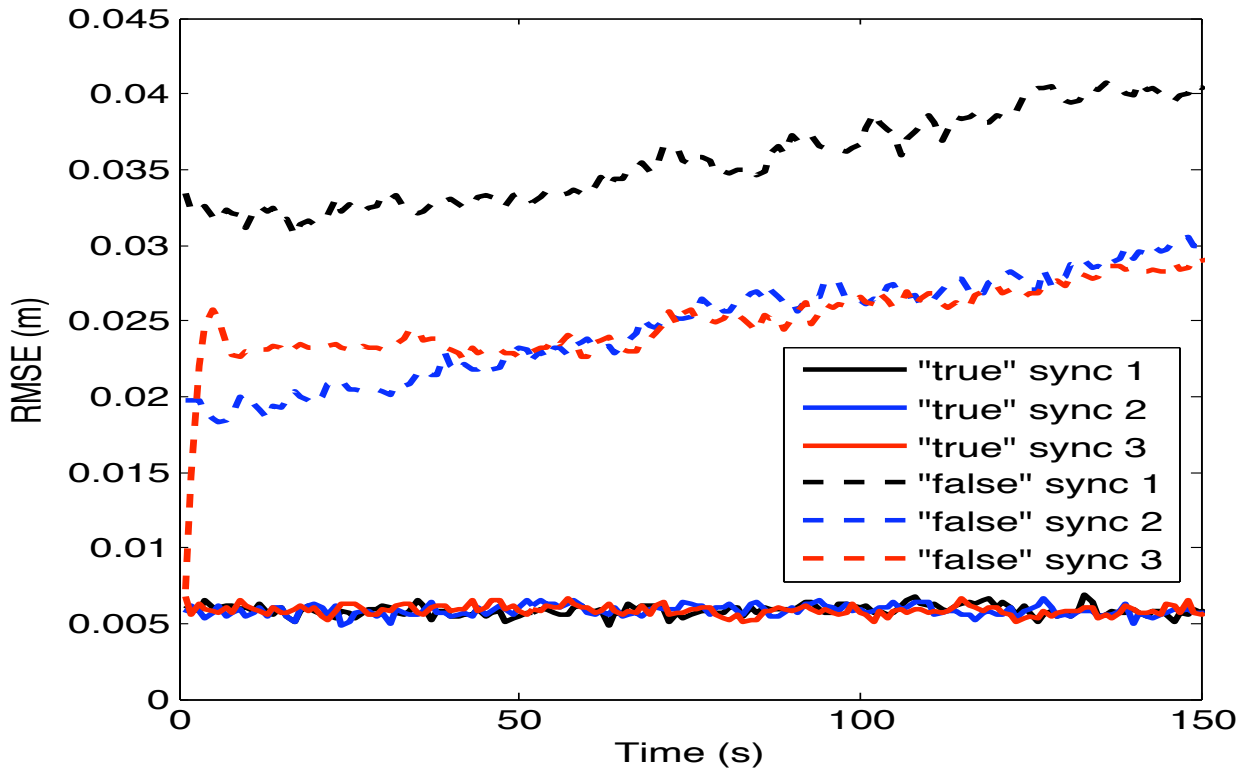


**Figure 4.3:** Baseline RMSE performance for different sets of measurement times

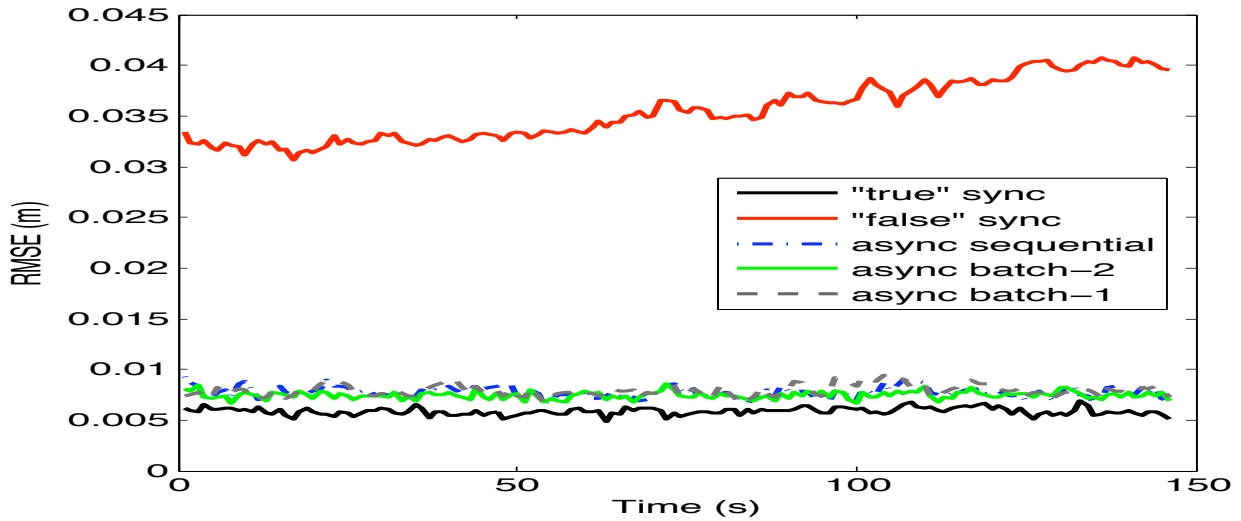


**Figure 4.4:** Performance for scenario 1 with asynchronous-compensation algorithms

equal run-time for the two algorithms) was monitored over various particle counts and is shown in figure 4.5. It is clear from the figure that the batch-1 method is the best in terms of computational load, and does not require much additional processing time compared to the "true" synchronous method. Also the sequential method outperforms the batch-2 algorithm only for a low number

of particles. It was initially believe that this result can likely be attributed to the sequential method's stronger dependency on particle count, with the Batch-2 method requiring roughly the same amount of "overhead" regardless of the particle count. Further consideration of the situation has brought this statement into doubt; since Batch-2 essentially creates $KNM$ "child" particles in each step to compute the integration. It is now believed that the higher computational load present in the sequential method is dominated by the additional amount of resampling which needs to be performed that is difficult to optimize, whereas the integration step in Batch-2 can be easily parallelized.

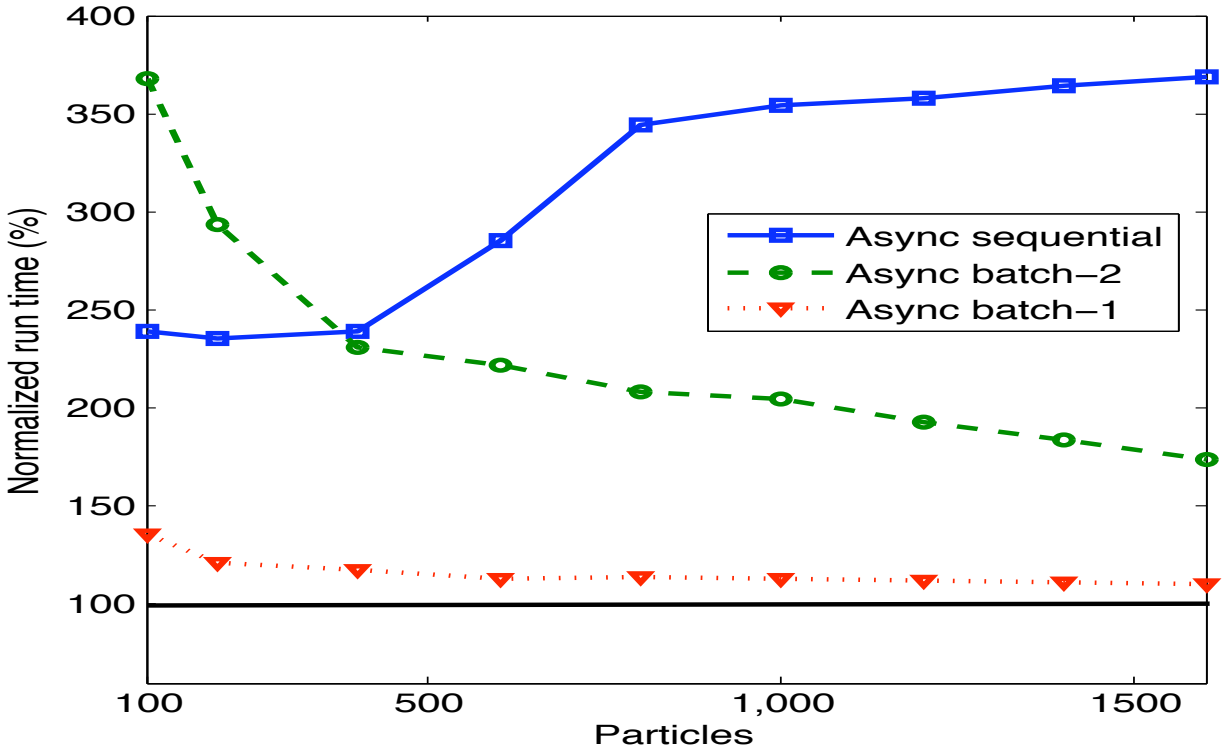


**Figure 4.5:** Normalized run-time as a function of the total number of particles

## 4.4 Concluding Remarks

It should be clear from the results presented that a number of effective solutions were developed in [3] to handle asynchronous measurements in the context considered. Revisitation of this topic has nonetheless revealed opportunities for future research development. Particularly, it is believed that the results obtained in the context of MASCOT relating to the RSS likelihood function can be applied to analytically evaluate the integral in (4.14) under specific restrictions, thereby eliminating the need for Monte Carlo integration as in the Batch-2 method. Another area of specific interest

lies in addressing the possibility of unknown and randomly varying base times for each agent which can certainly occur in a realistic tracking environment.

# 5

# Bayesian Performance Metrics

## 5.1 Overview

A reasonable and frequently employed construct for statistical inference is the MMSE estimator $\hat{\theta} = \phi(y)$: the estimate of $\theta$ based on $y$ which will yield, *on the average*, the minimum squared-error. Thus we seek this construct as the solution to our inference problem, though we may not always succeed due to mathematical intractability. In this case, sub-optimal methods are adopted which normally attempt to emulate $\phi(y)$ as closely as possible. Whether we can obtain $\phi(y)$ or not (and particularly when we cannot) it is highly desirable to "baseline" the problem; to develop a bound indicating the best possible performance we can hope to achieve in the problem given our known statistics. In fact, this bound may not guarantee our ability to attain this level of performance, but it *will guarantee that we can do no better*. Equipped with this performance bound gives us a relatively reliable indicator of how close some particular sub-optimal estimator we have developed is to being optimal; if we are very close, it may not be worth the time or resource to continue searching for improvements.

The value of developing performance bounds for statistical inference was recognized long ago and a large amount of work now exists which focuses on the establishment of bounds to all sorts of problems. In nearly all of the literature, there is a distinct boundary between classical performance bounds (which address inference of a non-random yet unknown parameter) and Bayesian bounds (that deal with random unknown parameters). The most well-known classical bound being the Cramer-Rao-Lower-Bound (CRLB) is formulated in [65]; the Bayesian analogue to this bound was developed by Van Trees [66]. These bounds can be "loose" in many problems meaning that they are not achievable by any estimator and as a result, tighter bounds can be found, with the tightest bound being the performance of the truly optimal estimator; many of these have been chronicled in [67]. It is important to mention here that these bounds relate to either the mean-square error (MSE) in the classical case, or the Bayesian mean-square error (BMSE), which are each valid for *a specific*

*class* of estimators; namely those which are unbiased in a classical sense or "average" unbiased in a Bayesian sense. As discussed in section 1.3 a fundamental challenge was encountered when investigating tracking performance of the algorithms discussed herein that demanded the author's attention. This challenge is of a general nature and relates not only to the contexts discussed within the thesis but in fact to any problem dealing with estimation of a random quantity. To elucidate the dilemma, suppose we are concerned with the estimation of some scalar random parameter $\theta_0$ that is drawn *once* and used as a fixed input parameter to draw $y$. We observe $y$ and wish to make inference about $\theta_0$; we have the following known statistics regarding the problem:

$$\theta_0 \sim f(\theta) \qquad y \sim f(y \mid \theta_0) \tag{5.1}$$

This problem can be interpreted and approached at from both a classical and Bayesian viewpoint. From the classical view, there is no importance as to how $\theta_0$ was originally generated; all we are concerned with is that it is now unknown but fixed, i.e., there is some *true* value taken on by $\theta_0$ we wish to determine. In this case, the MSE of any estimator, $\hat{\theta}_0$, developed will be bound by the classical CRLB; assuming the estimator is unbiased, i.e. $b(\theta_0) = \mathbb{E}_y\left[\left(\theta_0 - \hat{\theta}_0\right)\right] = 0$, its variance can be no less than the inverse Fisher Information, or:

$$MSE(\theta_0) = \mathbb{E}_y\left[\left(\theta_0 - \hat{\theta}_0\right)^2\right] \geq J_D(\theta_0)^{-1} \tag{5.2}$$

Note here that $\mathbb{E}_y$ denotes an expectation taken over the random measurements according to $f(y \mid \theta)$, $\mathbb{E}_\theta$ denotes an expectation taken over the random parameter according to the prior $f(\theta)$, and the following information terms are defined here which will be used frequently in what follows:

$$\begin{aligned} J_D(\theta) &= \mathbb{E}_y\left[\frac{-\partial^2 \log f(y \mid \theta)}{\partial \theta^2}\right] && \text{Fisher Information} \\ J_P &= \mathbb{E}_\theta\left[\frac{-\partial^2 \log f(\theta)}{\partial \theta^2}\right] && \text{Prior Information} \\ J_B &= \mathbb{E}_\theta\left[\mathbb{E}_y\left[\frac{-\partial^2 \log f(y,\theta)}{\partial \theta^2}\right]\right] && \text{Bayesian Information} \end{aligned}$$

Here $f(y,\theta) = f(y \mid \theta)f(\theta)$ is the joint density of the parameter/measurement pair; note only the Fisher Information term has a dependency of the actual value of $\theta$.

In a Bayesian sense, $\theta_0$ is random and our prior belief of likely values it may assume is embodied within $f(\theta)$. We can again form an estimator $\hat{\theta}_0$ that takes into account this prior knowledge and the BMSE is bounded by the Bayesian MSE; assuming the estimator is "average unbiased", or $\int b(\theta_0)f(\theta)d\theta = 0$, then its BMSE can be no less than the Bayesian Information, or:

$$BMSE = \mathbb{E}_\theta\left[MSE(\theta_0)\right] \geq J_B^{-1} \tag{5.3}$$

Now, the disconnect can be made clear: in the classical sense we can bound performance of $\hat{\theta}_0$ and this bound will depend on the *true* value of $\theta_0$. But there is no meaning for prior information in this

bound according to (5.2). In the Bayesian sense, the bound indeed depends on the prior information $f(\theta)$, yet there is now no dependency on *particular realizations* of $\theta_0$ (hence the absence of any argument from $BMSE$); any information regarding performance (averaged over the measurements alone) at specific parameter realizations has been averaged out in the process of forming the BMSE. Yet there is a clear interest in having information pertaining to performance at specific realizations of $\theta_0$, particularly in a target tracking setting which is inherently based on the incorporation of prior knowledge concerning the target dynamics. It will be shown how this information can indeed be obtained by recognizing that there exists a quantifiable link between a Bayesian environment and a classical one; optimal Bayesian estimation is equivalent to classical biased estimation governed by the form of the prior knowledge.

## 5.2 Introduction to the Bayesian-Bias Connection

### 5.2.1 Related Work: The Optimal Bias Function

The connection between a biased estimator of a deterministic parameter and Bayesian estimation when this parameter is subsequently assumed random with respect to a given prior distribution has been recognized before. It was shown in [68] and later extended in [69] that an alternate bound for the Bayesian MSE can be developed by finding the bias function, $b(\theta)$, which minimizes the expectation of the biased Cramer-Rao-Lower-Bound (CRLB),

$$CRLB\left[\theta, b(\theta), \frac{\partial b(\theta)}{\partial \theta}\right] = \frac{\left[1+\frac{\partial b(\theta)}{\partial \theta}\right]^2}{J_D(\theta)} + b(\theta)^2, \tag{5.4}$$

over the prior distribution of $\theta$. Equation (5.4) is a bound on the MSE of any estimator $\hat{\theta}$ of $\theta$ that is assumed deterministic but unknown, and where $\hat{\theta}$ has specific bias function at $\theta$, $b(\theta)$, specifically:

$$\mathbb{E}_y\left[\left(\theta-\hat{\theta}\right)^2\right] \geq CRLB\left[\theta, b(\theta), \frac{\partial b(\theta)}{\partial \theta}\right]$$
$$\text{when} \qquad \mathbb{E}_y\left[\left(\theta-\hat{\theta}\right)\right] = b(\theta) \tag{5.5}$$

Observe that the CRLB is denoted with arguments in (5.4) to clarify that it is in fact a *functional* of the specific estimator bias function and its first derivative with respect to $\theta$. This classical bound can be applied to a random parameter by treating each value of $\theta$ in the argument of (5.4) as a *realization* of the parameter. The bias function $b(\theta)$ is then interpreted as a conditional bias given that $\theta$ has realized that particular value, i.e.,

$$b(\theta) = \int (\hat{\theta}-\theta) f(y|\theta) dy \tag{5.6}$$

Any estimator $\hat{\theta}$ of a random parameter $\theta$ from measurements $y$ with this bias function then obeys the following inequality on Bayesian MSE:

$$\begin{aligned} BMSE &= \iint (\hat{\theta}(y) - \theta)^2 f(y|\theta) f(\theta) dy d\theta \\ &\geq \int CRLB\left[\theta, b(\theta), \frac{\partial b(\theta)}{\partial \theta}\right] f(\theta) d\theta \end{aligned} \tag{5.7}$$

Thus any estimator will have a BMSE that is lower-bounded as:

$$BMSE \geq \inf_{\mathbf{b} \in \mathbb{H}^1} \int CRLB\left[\theta, b(\theta), \frac{\partial b(\theta)}{\partial \theta}\right] f(\theta) d\theta \tag{5.8}$$

where $\mathbb{H}^1$ is the first-order Sobolev space, or the space of all weakly-differentiable functions in $\mathbb{L}^2$, the space of all square-integrable functions. The function-space is carefully defined this way to ensure the existence of a unique minimizer $b^\star(\theta)$; see pages 3-4 of [69] for further details.

Developing a lower bound on the BMSE can thus be seen as a functional optimization problem whose solution yields a differential equation describing the "optimal bias function", along with the resulting minimum BMSE, coined the "Optimal Bias Bound". This problem was originally solved in [68] for the specific case of a single scalar parameter $\theta$ with bounded support over $(\theta_0, \theta_1)$ and boundary conditions $\left[\frac{\partial b(\theta)}{\partial \theta}|_{\theta=\theta_0} = \frac{\partial b(\theta)}{\partial \theta}|_{\theta=\theta_1} = -1\right]$ (which are naturally implied by the boundedness assumption) and is restated here for convenience [69]:

$$J_D(\theta) b(\theta) = \frac{\partial^2 b(\theta)}{\partial^2 \theta} + \left(1 + \frac{\partial b(\theta)}{\partial \theta}\right)\left(\frac{\partial \log f(\theta)}{\partial \theta} - \frac{\partial \log J_D(\theta)}{\partial \theta}\right) \tag{5.9}$$

A function $b^\star(\theta)$ satisfying (5.9) and the aforementioned assumptions is then a unique minimizer of (5.8). In general, no estimator may exist possessing the optimal bias, however this still represents a bound on the BMSE viewed in terms of a specific form of the deterministic bias function.

While the original solution presented in [69] rested on the assumption of bounded support over $\theta$, it will now be shown that (5.9) describes the optimal bias even in cases with unbounded support as long as $J_D(\theta)$ is nonzero for all $\theta$ and $f(\theta)$ vanishes as $\theta \to \pm\infty$. To show this is true, let us first make the definition:

$$Z\left[b(\theta), \frac{\partial b(\theta)}{\partial \theta}\right] = \int_{-\infty}^{\infty} CRLB\left[\theta, b(\theta), \frac{\partial b(\theta)}{\partial \theta}\right] f(\theta) d\theta \tag{5.10}$$

and find the function $b(\theta)$ which minimizes this functional. $Z[b, b']$ will be minimized when its first variation $\delta Z[b, b'] = 0$ (see [70] for a review of the theory of variational calculus). First letting $F(\theta, b(\theta), b'(\theta)) = CRLB\left[\theta, b(\theta), \frac{\partial b(\theta)}{\partial \theta}\right] f(\theta)$ and using a Taylor series approximation to compute $\delta Z$ about $b(\theta)$ yields:

$$\triangle Z\left[b(\theta), b'(\theta)\right] = \epsilon \int_{-\infty}^{\infty} \frac{\partial F}{\partial b} v(\theta) d\theta + \epsilon \int_{-\infty}^{\infty} \frac{\partial F}{\partial b'} v'(\theta) d\theta \tag{5.11}$$

Where $v(\theta)$ is the "direction" of variation, $\triangle Z$ is the principle linear part of $\delta Z$ and $\triangle Z \rightarrow \delta Z$ as the constant $\epsilon \rightarrow 0$. A minimizer, $b^{\star}(\theta)$ of $Z$ must satisfy $\triangle Z \left[ b^{\star}(\theta), \frac{\partial b^{\star}(\theta)}{\partial \theta} \right] = 0$ for any $v(\theta)$. Then making use of the basic product derivative property to express the second term in the RHS of (5.11) as:

$$\int_{-\infty}^{\infty} \frac{\partial F}{\partial b'} v'(\theta) d\theta = \left[ \frac{\partial F}{\partial b'} v(\theta) \right]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \frac{\partial}{\partial \theta} \left( \frac{\partial F}{\partial b'} \right) v(\theta) d\theta \tag{5.12}$$

Substituting (5.12) into (5.11) and expanding F produces:

$$\triangle Z = \epsilon \left( \int_{-\infty}^{\infty} 2 \left[ b(\theta) f(\theta) - \frac{\partial}{d\theta} \left( \frac{\left[ 1 + b'(\theta) \right] f(\theta)}{J_D(\theta)} \right) \right] v(\theta) d\theta + \left[ 2 \frac{\left[ 1 + b'(\theta) \right] f(\theta)}{J_D(\theta)} v(\theta) \right]_{-\infty}^{\infty} \right)$$

Applying the stated assumptions forces to zero the second term in the RHS of (5.13). Since $\triangle Z = 0$ at its minimum, one arrives arrive at the final result by expanding the first RHS term in (5.13) and noting that the integrand must vanish for any $v(\theta)$:

$$\frac{\left[ \frac{\partial^2 b(\theta)}{\partial \theta^2} f(\theta) + \left( 1 + \frac{\partial b(\theta}{\partial \theta} \right) \frac{\partial f(\theta)}{\partial \theta} \right] J_D(\theta) - \left[ 1 + \frac{\partial b(\theta)}{\partial \theta} \right] f(\theta) \frac{\partial J_D(\theta)}{\partial \theta}}{J_D(\theta)^2} =$$

$$b(\theta) f(\theta) - \frac{f(\theta)}{J_D(\theta)} \left( \frac{\partial^2 b(\theta)}{\partial \theta^2} + \left( 1 + \frac{\partial b(\theta)}{\partial \theta} \right) \left( \frac{\partial \log f(\theta)}{\partial \theta} - \frac{\partial \log J_D(\theta)}{\partial \theta} \right) \right) = 0 \tag{5.13}$$

which is equivalent to (5.9)

### 5.2.2 Generalization of the Optimal Bias Function

Generalizing the concept of an "optimal bias" function can provide one with deeper insight into the relationship between the optimal Bayesian estimator and biasedness in a deterministic estimation setting. Assume we have some deterministic bound $\mathbb{B}$ so that,

$$MSE \left[ \tilde{\theta} \mid \theta \right] \geq \mathbb{B} \left[ \theta, b(\theta) \right]$$

for any estimator $\tilde{\theta}(y)$ with bias function $b(\theta)$ (this notation means the MSE of the estimator $\tilde{\theta}$ conditioned on the parameter realization $\theta$). Let us also make the assumption that there exists an optimal bias function, unique up to probability measure zero on the space of $\theta$, $b^{\star}(\theta) \in \mathbb{H}^1$ that is a minimizer of the expectation over $f(\theta)$ of the bound functional $\mathbb{B}$:

$$\int \mathbb{B} \left[ \theta, b(\theta) \right] f(\theta) d\theta \geq \int \mathbb{B} \left[ \theta, b^{\star}(\theta) \right] f(\theta) d\theta \triangleq \mathbb{BB} \tag{5.14}$$

The right-hand side of (5.14) represents a bound on the BMSE, so we necessarily have:

$$BMSE \left[ \hat{\theta}(y) \right] \geq \mathbb{BB} \tag{5.15}$$

for any estimator $\hat{\theta}(y)$ possessing *any* bias function. It is asserted that if in fact this bound is achievable, then $b^\star(\theta)$ is possessed, almost everywhere, by at least the optimal Bayesian estimator, $\tilde{\theta}_B(y)$. One also has that:

$$MSE\left[\tilde{\theta}_B \mid \theta\right] = \mathbb{B}\left[\theta, b^\star(\theta)\right] \qquad \text{almost everywhere on } \theta \tag{5.16}$$

To show this, let us first establish that $b^\star(\theta)$ must be possessed by at least one estimator if $\mathbb{BB}$ is achievable. If this were not the case, then by the uniqueness assumption of $b^\star(\theta)$ we would have the strict inequality:

$$\int \mathbb{B}\left[\theta, b(\theta)\right] f(\theta) d\theta > \int \mathbb{B}\left[\theta, b^\star(\theta)\right] f(\theta) d\theta \tag{5.17}$$

for any bias $b(\theta)$ that is possessed by some estimator. This implies $\mathbb{BB}$ is not achievable, and is a contradiction. The optimal Bayesian estimator must then have bias $b^\star(\theta)$ for if it did not, it would not achieve $\mathbb{BB}$, which is again a contradiction. We then have for the optimal Bayesian estimator that:

$$\int MSE\left[\tilde{\theta}_B|\theta\right] f(\theta) d\theta = \int \mathbb{B}\left[\theta, b^\star(\theta)\right] f(\theta) d\theta \tag{5.18}$$

which implies (5.16) since $f(\theta) \geq 0$ and $\mathbb{B}$ is a bound on any estimator with bias $b^\star(\theta)$.

This suggests the "deterministic" interpretation of an optimal Bayesian estimator, $\tilde{\theta}_B(y)$, as that estimator with bias function $b_B(\theta)$ having uniformly minimum mean square error almost everywhere under $f(\theta)$; out of all estimators with bias $b_B(\theta)$, $MSE\left[\hat{\theta}_B \mid \theta\right]$ is minimum for all $\theta$. The actual bias $b_B(\theta)$ is not arbitrary, but is such that the greatest lower bound (achievable) on the Bayesian MSE under $f(\theta)$ is minimized. In fact, this seems to suggest an approach to evaluating performance of an optimal Bayesian estimator at specific parameter realizations without ever explicitly constructing the estimator itself. If there is some Bayesian bound constructed as done in the preceding discussion, based on an underlying deterministic bound and a prior distribution which is known to be achievable for a specific problem, then it is theoretically possible to determine the optimal bias function, automatically yielding the optimal realization-specific MSE. Unfortunately, this is generally not feasible, since there are few problems actually possessing currently known achievable and easily computable bounds. Nonetheless, specific problems have indeed been found for which tighter bounds are achievable, for example see [71], and it is expected that one may directly apply the theory developed here to these special cases. [72] offers a promising technique based on Reproducing-Kernel-Hilbert-Space-Methods (RKHS) that perhaps can be used in conjunction with this towards specific application of the preceding development in a less restricted set of problems. This concept will be explored with the CRLB and Bayesian Cramer Rao Bound (BCRLB), which are both known to be achievable for a restricted class of estimation problems.

## 5.3 The Bias Connection in Estimation Problems of a Single Random Parameter

### 5.3.1 Theoretical Results

The fact that the bias function provides the connection between classical and Bayesian estimation is most apparent in the case of *efficient estimation* problems; ones in which there exists an estimator which achieves the BCRLB.

Assuming that the standard regularity conditions hold, namely that the probability density functions governing the problem are twice differentiable, and:

$$\mathrm{E}\left[\frac{\partial \log f(y|\theta)}{\partial \theta}\right] = \mathrm{E}\left[\frac{\partial \log f(\theta)}{\partial \theta}\right] = 0$$

The BCRLB then reads:

$$BMSE\left[\hat{\theta}\right] \geq BCRLB = J_B^{-1} \tag{5.19}$$

An efficient estimator which achieves the BCRLB exists if and only if there exists a function $\hat{\theta}(y)$ for which the following holds:

$$\frac{\partial \log f\left(y|\theta\right)}{\partial \theta} + \frac{\partial \log f\left(\theta\right)}{\partial \theta} = K(\hat{\theta}(y) - \theta) \tag{5.20}$$

where K is a constant and $\hat{\theta}(y)$ does not depend on $\theta$. The optimal Bayesian estimator is then $\hat{\theta}(y)$ and $K = J_B$.

Using this condition for efficiency under the BCRLB, an important result can be stated: Any efficient Bayesian estimator of $\theta$ which achieves the BCRLB is *equivalent to a classical biased estimator* with bias function $b^{\star}(\theta) = J_B^{-1}\left(\frac{\partial \log f(\theta)}{\partial \theta}\right)$. Furthermore, if the BCRLB is indeed achievable, the MSE of any estimator with this bias *at a given realization* of $\theta$ is bounded by:

$$MSE\left[\hat{\theta} \mid \theta\right] \geq J_D(\theta)^{-1}\left(1 + J_B^{-1}\frac{\partial^2 \log f(\theta)}{\partial \theta^2}\right)^2 + \left(J_B^{-1}\left[\frac{\partial \log f(\theta)}{\partial \theta}\right]\right)^2 \tag{5.21}$$

This can be shown simply by referring to (5.20) and directly evaluating what $b^{\star}(\theta)$ must be. We know that if this condition holds, then $\hat{\theta}(y)$ is the optimal Bayesian estimator. Thus:

$$\begin{aligned} b^{\star}(\theta) &= \int (\hat{\theta}(y) - \theta) f(y|\theta) d\theta \\ &= \int J_B^{-1}\left(\frac{\partial \log f\left(y|\theta\right)}{\partial \theta} + \frac{\partial \log f\left(\theta\right)}{\partial \theta}\right) f(y|\theta) d\theta \\ &= J_B^{-1}\left(\int \frac{\partial \log f\left(y|\theta\right)}{\partial \theta} f\left(y|\theta\right) d\theta + \frac{\partial \log f\left(\theta\right)}{\partial \theta}\right) \\ &= J_B^{-1}\left(\frac{\partial \log f(\theta)}{\partial \theta}\right) \end{aligned}$$

Where the last expression follows from the assumed regularity condition $\mathrm{E}_y\left[\frac{\partial \log f\left(y|\theta\right)}{\partial \theta}\right] = 0$ and the independence of $\frac{\partial \log f(\theta)}{\partial \theta}$ on $y$. As we now have an expression for $b^\star(\theta)$ we can simply use this in (5.4) to arrive at (5.21). Note this has a similar form to the earlier discussion involving a generalized bound $\mathbb{B}$.

Let us further solidify these notions by applying this to the simplest possible scenario, where $\theta \sim \mathcal{N}(A, \sigma_A)$ with A known, $y \sim \mathcal{N}(\theta, \sigma_y)$ and we wish to estimate $\theta$. Both the Fisher and Prior information terms $J_D(\theta)$ and $J_P$ are constant in this case and we have $BCRLB = \frac{\sigma_A^2 \sigma_y^2}{\sigma_A^2 + \sigma_y^2}$. Since $\frac{\partial \log f(\theta)}{\partial \theta} = \frac{-1}{\sigma_A^2}(\theta - A)$, the optimal bias function is:

$$b^\star(\theta) = \frac{-\sigma_y^2}{\sigma_y^2 + \sigma_A^2}(\theta - A) \tag{5.22}$$

with realization-specific MSE bounded as:

$$\begin{aligned} MSE\left[\hat{\theta} \mid \theta\right] \geq & \\ & \sigma_y^2 \left[1 + \left(\frac{\sigma_A^2 \sigma_y^2}{\sigma_A^2 + \sigma_y^2}\right)\left(\frac{-1}{\sigma_A^2}\right)\right]^2 + \left[\frac{-\sigma_y^2}{\sigma_y^2 + \sigma_A^2}(\theta - A)\right]^2 \\ & = \frac{\sigma_y^2}{(\sigma_A^2 + \sigma_y^2)^2}\left[\sigma_A^4 + \sigma_y^2(\theta - A)^2\right] \end{aligned} \tag{5.23}$$

Letting $\alpha = \frac{\sigma_A^2}{\sigma_A^2 + \sigma_y^2}$ we have:

$$MSE\left[\hat{\theta} \mid \theta\right] \geq \alpha^2 \sigma_y^2 + (1 - \alpha)^2(\theta - A)^2 \tag{5.24}$$

This is a well-known result and is indeed the MSE at specific realizations of $\theta$ of the posterior mean MMSE. A "risk-reward" tradeoff [73] can clearly be seen in (5.24); the MSE will be lower than the unbiased CRLB in a neighborhood around $\theta = A$ but increases without bound as $\theta$ is realized farther from this interval. This trade-off holds in general; we can find biased estimators that do better than the CRLB at localized parameter regions but which may yield significantly worse performance at other regions. See [74] and [75] for analysis of this bias-variance tradoff in a non-Bayesian setting. What can also be easily shown in this case is that $b^\star(\theta)$ is indeed a solution to the differential equation in (5.9). We have:

$$\left(\frac{1}{\sigma_y^2}\right)\left(\frac{-\sigma_y^2}{\sigma_y^2 + \sigma_A^2}(\theta - A)\right) = (0) + \left(1 + \left(\frac{-\sigma_y^2}{\sigma_y^2 + \sigma_A^2}\right)\right)\left(\left(\frac{-1}{\sigma_A^2}(\theta - A)\right) - (0)\right) \tag{5.25}$$

which is clearly true, confirming $b^\star(\theta)$ is a solution.

While simple, the preceding example is instructive as it solidifies the notion of a generalized

optimal bias bound $\mathbb{B}$ mentioned earlier. Although the BCRB was not initially constructed from a deterministic bias-specific bound, it is easily seen that it coincides with the expectation over the prior of the bound in (5.21) when it is achievable. To this end we first make the following observation regarding $J_D(\theta)$ which follows directly from (5.20):

$$J_D(\theta) = \int_{-\infty}^{\infty} \frac{-\partial^2 \log f(y|\theta)}{\partial\theta^2} f(y|\theta)dy = \int_{-\infty}^{\infty} -\frac{\partial}{\partial\theta}\left\{K\left(\hat{\theta}(y) - \theta\right) - \frac{\partial \log f(\theta)}{\partial\theta}\right\} f(y \mid \theta)\, dy$$
$$= J_B + \frac{\partial^2 \log f(\theta)}{\partial\theta^2}$$

As such, (5.21) can be rewritten as:

$$MSE\left[\hat{\theta} \mid \theta\right] \geq \frac{\left(1 + J_B^{-1}\frac{\partial^2 \log f(\theta)}{\partial\theta^2}\right)^2}{\left(J_B + \frac{\partial^2 \log f(\theta)}{\partial\theta^2}\right)} + \left(J_B^{-1}\left[\frac{\partial \log f(\theta)}{\partial\theta}\right]\right)^2$$
$$= J_B^{-1} + J_B^{-2}\frac{\partial^2 \log f(\theta)}{\partial\theta^2} + J_B^{-2}\left(\frac{\partial \log f(\theta)}{\partial\theta}\right)^2 \quad (5.26)$$

Taking an expectation over the prior yields:

$$J_B^{-1} - J_B^{-2}J_P + J_B^{-2}\int_{-\infty}^{\infty}\left(\frac{\partial \log f(\theta)}{\partial\theta}\right)^2 f(\theta)\, d\theta \quad (5.27)$$

Yet the regularity assumption implies the third term is equal to $J_B^{-2}J_P$, leaving us with a single term for (5.27), $J_B^{-1}$. Based on this observation alone, we could have deduced that the optimal Bayesian estimator must achieve (5.21) for all $\theta$, since in this context, the expectation over (5.21) is indeed an achievable Bayesian bound constructed from a corresponding bias-specific deterministic bound, and therefore obeys (5.16).

An interesting bound on the estimation of a deterministic parameter while still assuming a prior distribution and treating the parameter as if it were random, was presented in [76]. This bound can also be interpreted as the minimum MSE of any estimator of a random parameter at specific realizations. It is restated here:

$$MSE\left[\hat{\theta} \mid \theta\right] \geq \frac{\left(1 + \frac{\partial \log(b(\theta)f(\theta))}{\partial\theta}b(\theta)\right)^2}{-\mathrm{E}_y\left[\frac{\partial^2 \log f(y|\theta)}{\partial\theta^2}\right] + \frac{\partial^2 \log f(\theta)}{\partial\theta^2}} \quad (5.28)$$

This bound differs from that of (5.21) in that it is for an arbitrary bias, not related to Bayesian optimality, and does not require the estimation problem to be efficient. It can be shown that it is naturally deduced from the inequality:

$$\int\left(\hat{\theta}(y) - \theta\right)^2 f(y|\theta)dy \geq \frac{\left(\int\left(\hat{\theta}(y) - \theta\right)\frac{\partial \log f(y,\theta)}{\partial\theta}f(y|\theta)dy\right)^2}{\int\left(\frac{\partial \log f(y,\theta)}{\partial\theta}\right)^2 f(y|\theta)dy} \quad (5.29)$$

for any estimator $\hat{\theta}(y)$. This leads us to an alternative expression for the optimal realization-specific MSE in an efficient problem. It is reminded here that the symbol $\hat{\theta}_B$ is used to denote the optimal Bayesian estimator, in contrast to the previous use of $\hat{\theta}$ which denotes any estimator of $\theta$. Than, any efficient estimator of $\theta$ which achieves the BCRLB has MSE at a particular realization of $\theta$ equal to:

$$MSE\left[\hat{\theta}_B \mid \theta\right] = \frac{\left(\frac{\partial \log f(\theta)}{\partial \theta}\right)^2 + J_D(\theta)}{\mathrm{E}_\theta\left[\left(\frac{\partial \log f(\theta)}{\partial \theta}\right)^2 + J_D(\theta)\right]^2} \tag{5.30}$$

This can be shown by noting an efficient estimator must satisfy (5.20) which is substituted for $\hat{\theta}(y)$ in (5.29), and is now an equality. We then arrive at the expression:

$$MSE\left[\hat{\theta}_B \mid \theta\right] = \left(J_B^{-1}\right)^2 \int \left(\frac{\partial \log f(y,\theta)}{\partial \theta}\right)^2 f(y|\theta) dy \tag{5.31}$$

Based on what was shown in [76]:

$$\mathrm{E}_y\left[\left(\frac{\partial \log f(y,\theta)}{\partial \theta}\right)^2\right] = \left(\frac{\partial \log f(\theta)}{\partial \theta}\right)^2 - \mathrm{E}_y\left[\frac{\partial^2 \log f(y|\theta)}{\partial \theta^2}\right] = \left(\frac{\partial \log f(\theta)}{\partial \theta}\right)^2 + J_D(\theta)$$

Thus we have:

$$MSE\left[\hat{\theta}_B \mid \theta\right] = \left(J_B^{-1}\right)^2 \left[\left(\frac{\partial \log f(\theta)}{\partial \theta}\right)^2 + J_D(\theta)\right]$$

Expanding the form of $J_B^{-1}$ in this expression yields the desired result, which is somewhat surprising yet intuitive. It appears to tell us that in the case of Bayesian efficiency, while having more information present in the data ( higher $J_D(\theta)$ ) will improve our overall minimum BMSE, $MSE\left[\hat{\theta}_B \mid \theta\right]$ at "unexpected" realizations with low $f(\theta)$ may be relatively worse if $J_D(\theta)$ is higher; the data "more strongly contradicts" our prior beliefs. To clarify, the following argument is provided: it will be seen shortly how the optimal bias tends to be larger at values of $\theta$ with low $f(\theta)$. Thus at these "outlier" realizations, the optimal estimator will "favor" higher $f(\theta)$-values in spite of what is indicated by the data. Essentially, *one cannot reap the full benefit of more informative data at outlier realizations due to the bias.*

These results can also be extended to estimation of a function of the parameter $\theta$ where efficiency can indeed be achieved whenever the posterior distribution has *maximum-entropy* under a specific set of constraints. Let us now consider an estimator $\hat{\theta}(y)$ of the function $g(\theta)$ of a random parameter $\theta$. The BCRLB in this case takes on the form:

$$BMSE \geq \frac{\mathrm{E}_\theta\left[\frac{\partial g(\theta)}{\partial \theta}\right]^2}{J_B^{-1}} \tag{5.32}$$

The bias function in this case is defined as:

$$b(\theta) = \int \left(\hat{\theta} - g(\theta)\right) f(y|\theta) dy \tag{5.33}$$

Then it can be stated that any efficient Bayesian estimator of $g(\theta)$ which achieves the BCRLB in (5.32) is equivalent to a classical biased estimator with bias function

$$b^*(\theta) = J_B^{-1} \mathrm{E}_\theta \left[\frac{\partial g(\theta)}{\partial \theta}\right] \left(\frac{\partial \log f(\theta)}{\partial \theta}\right) \tag{5.34}$$

Proving this is true proceeds identically to what was just shown in estimating $\theta$ itself when (5.32) is used in place of (5.19) and the condition for efficiency (5.20) is slightly modified to read:

$$\frac{\partial \log f(y|\theta)}{\partial \theta} + \frac{\partial \log f(\theta)}{\partial \theta} = K(\hat{\theta}(y) - g(\theta)) \tag{5.35}$$

for some function $\hat{\theta}(y)$ which does not depend on $\theta$, and $K = \frac{J_B}{\mathrm{E}_\theta\left[\frac{\partial g(\theta)}{\partial \theta}\right]}$.

We can again formulate a bound on the minimum MSE with this given bias at a particular realization of $\theta$ similar to that for estimation of $\theta$. However, instead an alternate form with a similar theme as in [76] will be developed. Let us start by extending (5.28) to estimation of $g(\theta)$ as follows:

$$f(\theta) \int \left(\hat{\theta}(y) - g(\theta)\right) f(y|\theta) = b(\theta) f(\theta) \tag{5.36}$$

Differentiating with respect to $\theta$ we have:

$$\int \left(\hat{\theta}(y) - g(\theta)\right) \frac{\partial \log f(y,\theta)}{\partial \theta} f(y|\theta) dy - \frac{\partial g(\theta)}{\partial \theta} f(\theta) = \frac{\partial b(\theta)}{\partial \theta} + \frac{\partial g(\theta)}{\partial \theta} + b(\theta) \frac{\partial \log f(\theta)}{\partial \theta} \tag{5.37}$$

Applying the Cauchy-Schwarz inequality this becomes:

$$\int \left(\hat{\theta}(y) - g(\theta)\right)^2 f(y|\theta) dy \geq \frac{\left(\frac{\partial b(\theta)}{\partial \theta} + \frac{\partial g(\theta)}{\partial \theta} + b(\theta) \frac{\partial \log f(\theta)}{\partial \theta}\right)^2}{\int \left(\frac{\partial \log f(y,\theta)}{\partial \theta}\right)^2 f(y|\theta) dy} \tag{5.38}$$

This leads to the result: that any estimator of $g(\theta)$ possessing the bias function of the efficient Bayesian estimator which achieves the BCRLB in (5.32) has MSE bounded at a particular realization of $\theta$ as:

$$MSE(\theta) \geq \frac{\left(\frac{\mathrm{E}_\theta\left[\frac{\partial g(\theta)}{\partial \theta}\right]}{\mathrm{E}_\theta\left[\frac{\partial \log f(\theta)}{\partial \theta}^2 + J_D(\theta)\right]} \frac{\partial^2 f(\theta)}{\partial \theta^2} \frac{1}{f(\theta)} + \frac{\partial g(\theta)}{\partial \theta}\right)^2}{\left(\frac{\partial \log f(\theta)}{\partial \theta}\right)^2 + J_D(\theta)} \tag{5.39}$$

This can be seen by simple substitution of the expression for the optimal bias function given in (5.34) into (5.38). As in the case of estimation of $\theta$, this leads to the result that any efficient

estimator of $g(\theta)$ which achieves the BCRLB has MSE at a particular realization of $\theta$ equal to:

$$MSE(\theta) = \mathrm{E}_\theta\left[\frac{\partial g(\theta)}{\partial \theta}\right]^2 \frac{\left(\frac{\partial \log f(\theta)}{\partial \theta}\right)^2 + J_D(\theta)}{\mathrm{E}_\theta\left[\left(\frac{\partial \log f(\theta)}{\partial \theta}\right)^2 + J_D(\theta)\right]^2} \tag{5.40}$$

As done in the case of estimating $\theta$ directly, it should be clear that:

$$\int (\,\hat{\theta}(y) - g(\theta)\,)^2 f(y|\theta)dy \geq \frac{\left(\int\left(\hat{\theta}(y) - g(\theta)\right)\frac{\partial \log f(y,\theta)}{\partial \theta} f(y|\theta)dy\right)^2}{\int\left(\frac{\partial \log f(y,\theta)}{\partial \theta}\right)^2 f(y|\theta)dy} \tag{5.41}$$

for any estimator $\hat{\theta}(y)$ of $g(\theta)$. But if $\hat{\theta}(y)$ is Bayesian efficient, we know that (5.35) is satisfied. Straightforward substitution of this expression into (5.41) directly confirms the result. It can in fact be shown that for efficient estimation, the right hand side of (5.40) is equivalent to the one shown in (5.39).

### 5.3.2 Single Parameter Estimation Performance Analysis with the Bias Connection

Let us now consider an informative example of how these results can be applied to yield realization-specific performance information concerning random parameter estimation. Assume are given N i.i.d samples $y_1, y_2, ..y_N$ from a Gaussian distribution with known mean $\theta$ and parameterized by the precision $\tau = \frac{1}{\sigma^2}$. We assume a prior for $\tau$ as Gamma with parameters $(\alpha, \beta)$ and wish to estimate the variance $\frac{1}{\tau}$. Let us start by computing the score functions:

$$\frac{\partial \log f(y|\tau)}{\partial \tau} = \frac{\partial}{\partial \tau}\log\left[\frac{1}{(2\pi)^{\frac{N}{2}}}\tau^{\frac{N}{2}} e^{-\frac{1}{2}\tau\sum_{i=1}^{N}(y_i-\theta)^2}\right] = \frac{N}{2\tau} - \frac{1}{2}\sum_{i=1}^{N}(y_i-\theta)^2$$

$$\frac{\partial \log f(\tau)}{\partial \tau} = \frac{\partial}{\partial \tau}\log\left[\frac{\beta^\alpha}{\Gamma(\alpha)}\tau^{\alpha-1}e^{-\beta\tau}\right] = \frac{\alpha-1}{\tau} - \beta$$

So we then have,

$$\begin{aligned}\frac{\partial \log f(y|\tau)}{\partial \tau} + \frac{\partial \log f(\tau)}{\partial \tau} &= \frac{\alpha - 1 + \frac{N}{2}}{\tau} + \frac{N}{2\tau} - \frac{1}{2}\sum_{i=1}^{N}(y_i-\theta)^2 \\ &= -\left(\alpha - 1 + \frac{N}{2}\right)\left(\frac{\sum_{i=1}^{N}(y_i-\theta)^2 + \beta}{\alpha - 1 + \frac{N}{2}} - \frac{1}{\tau}\right)\end{aligned}$$

We can see that the condition for efficiency is satisfied and the optimal Bayesian estimator of $\frac{1}{\tau}$ is

$$\hat{\theta}(y) = \frac{\sum_{i=1}^{N}(y_i-\theta)^2 + \beta}{\alpha - 1 + \frac{N}{2}}$$

It can be shown that this is indeed the posterior mean MMSE estimator. Note that the posterior mean (which is always the optimal Bayesian estimator) has never explicitly entered our discussion; it was implicitly constructed through the underlying bounds alone. Next we evaluate the Fisher Information:

$$\begin{aligned} J_D(\tau) &= \mathrm{E}_y\left[\left(\frac{N}{2\tau} - \frac{1}{2}\sum_{i=1}^{N}(y_i-\theta)^2\right)^2\right] = \mathrm{E}_y\left[\frac{N^2}{4\tau^2} - \frac{N}{2\tau}\sum_{i=1}^{N}(y_i-\theta)^2 + \frac{1}{4}\left(\sum_{i=1}^{N}(y_i-\theta)^2\right)^2\right] \\ &= -\frac{N^2}{4\tau^2} + \frac{1}{4}\mathrm{E}_y\left[N\sum_{k=1}^{N}(y_k-\theta)^4 + (N^2-N)\sum\sum_{i\neq j}(y_i-\theta)^2(y_j-\theta)^2\right] \\ &= -\frac{N^2}{4\tau^2} + \frac{1}{4}\mathrm{E}_y\left[\frac{3N}{\tau^2} + \frac{N^2-N}{\tau^2}\right] = \frac{N}{2\tau^2} \end{aligned}$$

With expectation under the prior as:

$$\int J_D(\tau)f(\tau)d\tau = \frac{N}{2}\int_0^\infty \frac{\beta^\alpha}{\Gamma(\alpha)}\tau^{\alpha-3}e^{-\beta\tau}d\tau = \frac{N\beta^2}{2(\alpha-1)(\alpha-2)}$$

The Prior Information term is:

$$\begin{aligned} &\int \frac{\partial \log f(\tau)}{\partial\tau}^2 f(\tau)d\tau = \int_0^\infty\left(\frac{\alpha-1}{\tau}-\beta\right)^2\frac{\beta^\alpha}{\Gamma(\alpha)}\tau^{\alpha-1}e^{-\beta\tau}d\tau \\ &= \int_0^\infty(\alpha-1)^2\frac{\beta^\alpha}{\Gamma(\alpha)}\tau^{\alpha-3}e^{-\beta\tau}d\tau - 2\int_0^\infty(\alpha-1)\frac{\beta^\alpha}{\Gamma(\alpha)}\tau^{\alpha-2}e^{-\beta\tau}d\tau + \beta^2 \\ &= \frac{\beta^2(\alpha-1)}{\alpha-2} - (2\beta^2) + \beta^2 = \frac{\beta^2}{\alpha-2} \end{aligned}$$

The Bayesian Information is then the sum of the previous two terms:

$$J_B = \frac{N\beta^2}{2(\alpha-1)(\alpha-2)} + \frac{\beta^2}{\alpha-2} = \frac{\beta^2(N+2(\alpha-1))}{2(\alpha-1)(\alpha-2)}$$

and the BCRLB is:

$$\begin{aligned} BCRLB &= \frac{\mathrm{E}_y\left[\frac{\partial}{\partial\tau}\left(\frac{1}{\tau}\right)\right]^2}{J_B} = \left[-\int_0^\infty\frac{\beta^\alpha}{\Gamma(\alpha)}\tau^{\alpha-3}e^{-\beta\tau}d\tau\right]^2\frac{2(\alpha-1)(\alpha-2)}{\beta^2(N+2(\alpha-1))} \\ &= \frac{2\beta^2}{(\alpha-1)(\alpha-2)(N+2(\alpha-1))} \end{aligned}$$

The bias of the efficient estimator is then:

$$b^{\star}(\tau) = \left[ \left( \frac{2\beta^2}{(\alpha-1)(\alpha-2)(N+2(\alpha-1))} \right) \left( \frac{-(\alpha-1)(\alpha-2)}{\beta^2} \right) \left( \frac{\alpha-1}{\tau} - \beta \right) \right]$$
$$= \frac{-2}{N+2(\alpha-1)} \left( \frac{\alpha-1}{\tau} - \beta \right)$$

Finally using (5.40) we calculate the realization-specific MSE of the Bayesian estimator:

$$MSE\left[\hat{\tau}_B \mid \tau\right] = \left[ \frac{-\beta^2}{(\alpha-1)(\alpha-2)} \right]^2 \frac{\left(\frac{\alpha-1}{\tau} - \beta\right)^2 + \frac{N}{2\tau^2}}{\left(\frac{\beta^2(N+2(\alpha-1))}{2(\alpha-1)(\alpha-2)}\right)^2} = \frac{4\left[\left(\frac{\alpha-1}{\tau} - \beta\right)^2 + \frac{N}{2\tau^2}\right]}{[N+2(\alpha-1)]^2}$$

The Bayes efficient bias function $b(\tau)$ for several values of $N$ with $\alpha = 10$ and $\beta = 3$ is shown in Figure 5.1 plotted alongside the prior density for reference. We can clearly see that the bias is minimum at the mode of the prior and decreases as N grows. It is not entirely clear why we end up with this particular functional form until we rescale the $\tau$-axis in terms of $\frac{1}{\tau}$. This is shown in figure 5.2 along with a rescaled version of the prior. From this we can see how the bias function acts to linearly "attract" estimates of $\frac{1}{\tau}$ back to its most likely value under the prior.

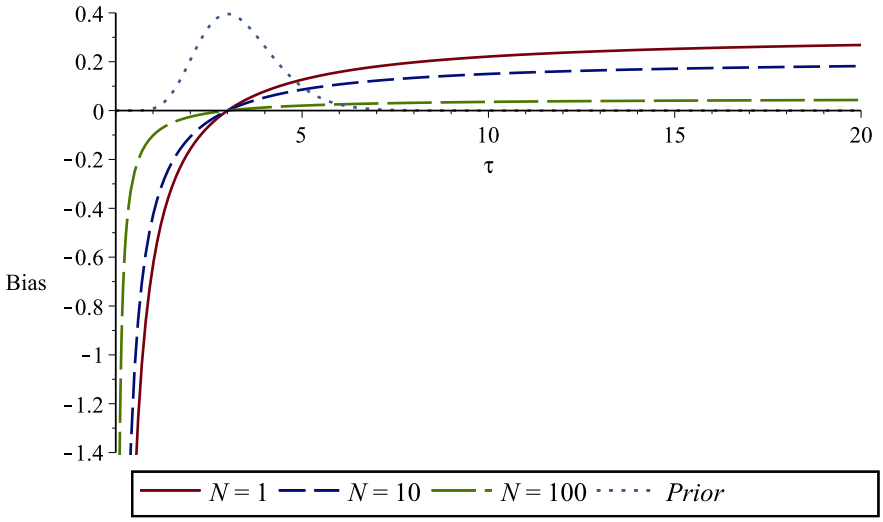


**Figure 5.1:** Efficient Bias Function $b(\tau)$ for estimation of $g(\tau) = \frac{1}{\tau}$ with $\alpha = 10$, $\beta = 10$

We next examine the realization-specific MSE shown in figures 5.3 and 5.4 for values of $N = 1$ and $N = 20$ respectively. Note in this case the plots are reparametrized to show the error in terms of $\sigma$ rather than $\tau$. It is clear the MSE is minimum and well below the CRLB at the prior mode. Also interesting to note in the case of $N = 1$ even as $\sigma$ departs to the right from the prior mode, the Bayes estimator MSE remains below the CRLB; this is not true for $N = 20$. In both cases as $\sigma \to 0$ the CRLB $\to 0$ while a finite error remains for the Bayes estimator.

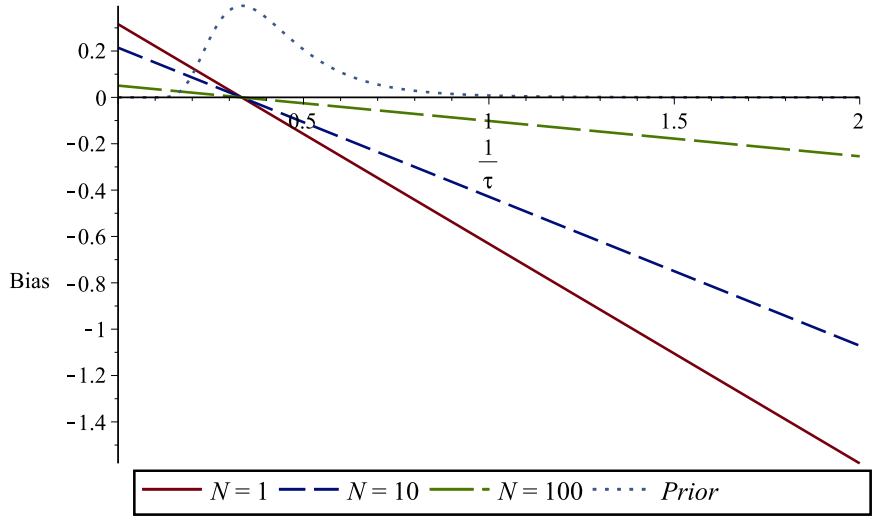


**Figure 5.2:** Efficient Bias Function $b(\tau)$ for estimation of $g(\tau) = \frac{1}{\tau}$ with $\alpha = 10$, $\beta = 10$ rescaled in terms of $\frac{1}{\tau}$

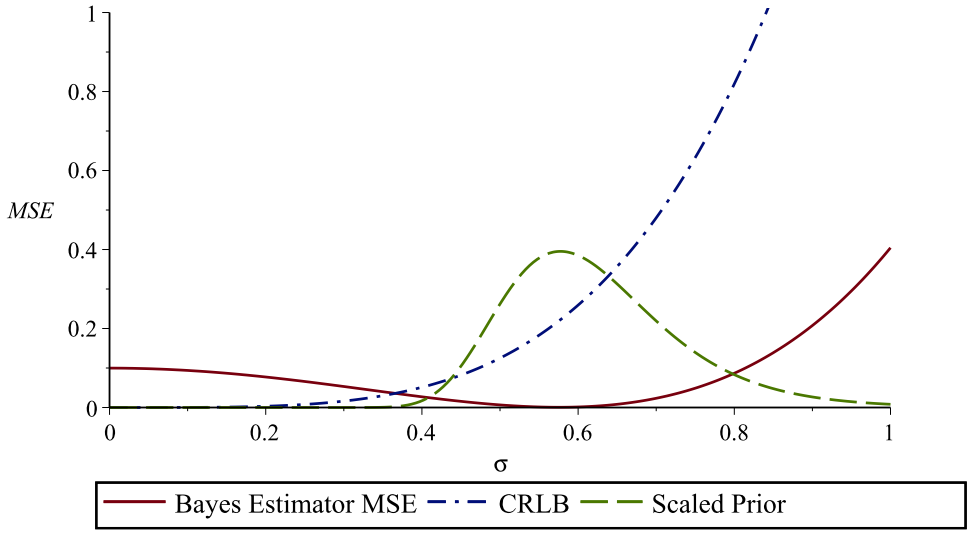


**Figure 5.3:** Realization specific MSE for estimation of $\frac{1}{\tau} = \sigma^2$ with N = 1 sample, $\alpha = 10$, $\beta = 3$

The ratio of realization-specific MSE to the CRLB is shown in figure 5.5. While it is obvious here the CRLB is approached asymptotically (view the 100K sample curve) larger sample sets below some threshold progressively diminish localized-superefficiency (herein defined as achieving an MSE lower than the unbiased CRLB, i.e. more efficient) of the Bayes estimator while still contributing to a larger error outside of the high $f(\tau)$ region. Existence of this type of behavior is dependent on the shape of the prior and it can be seen that this is not present with a less informative (higher variance) one, ($\alpha = 3$, $\beta = 3$) shown in figure 5.6.

We can take this one step further and examine the interaction between the hyperparameters,

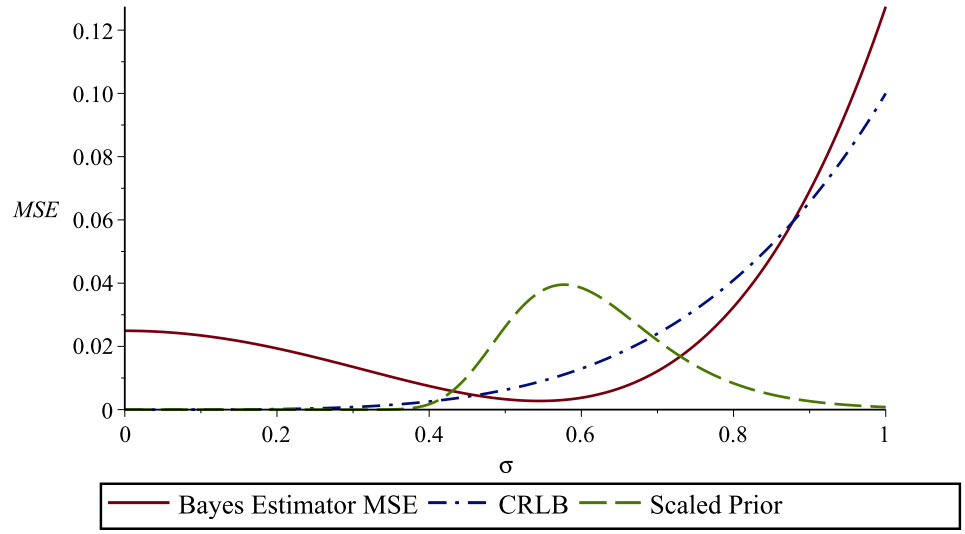


**Figure 5.4:** Realization specific MSE for estimation of $\frac{1}{\tau} = \sigma^2$ with N = 20 samples, $\alpha = 10$, $\beta = 3$

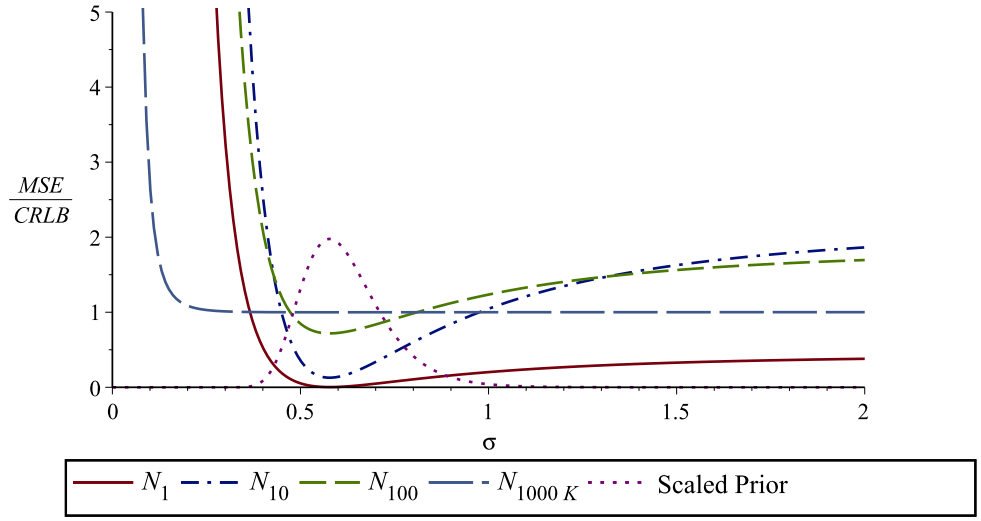


**Figure 5.5:** Ratio of $\frac{MSE}{CRLB}$ for various N, $\alpha = 10$, $\beta = 3$

sample size, and ratio of realization-specific MSE, $MSE\left[\hat{\tau}_B \mid \tau\right]$, to CRLB at key values of $\sigma$. To this end we first express the reparametrization of this ratio (this is arbitrarily done for convenience and the analysis could have easily been left in terms of $\tau$).

$$T(\sigma) \triangleq \frac{MSE\left[\hat{\tau}_B \mid \tau\right]}{CRLB(\sigma)} = \left(\frac{4\left(\left[\alpha-1\right]\sigma^2-\beta\right)^2+2N\sigma^4}{\left(N+2\alpha-2\right)^2}\right)\left(\frac{N}{2\sigma^4}\right)$$

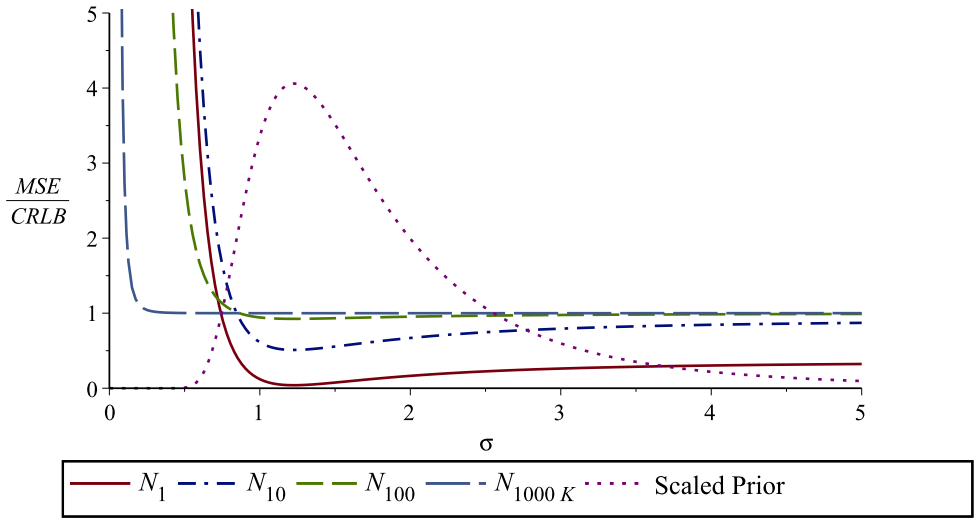


**Figure 5.6:** Ratio of $\frac{MSE}{CRLB}$ for various N, $\alpha = 3$, $\beta = 3$

Taking the derivative with respect to $\sigma$ in order to find its minimum we have:

$$\frac{\partial}{\partial\sigma}T(\sigma) = \left(\frac{16\left[(\alpha-1)\,\sigma^2-\beta\right](\alpha-1)\,\sigma+8N\sigma^3}{(N+2\alpha-2)^2}\right)\left(\frac{N}{2\sigma^4}\right)$$
$$-\left(\frac{4\left[(\alpha-1)\,\sigma^2-\beta\right]^2+2N\sigma^4}{(N+2\alpha-2)^2}\right)\left(\frac{2N}{\sigma^5}\right) = \frac{8N\beta\left((\alpha-1)\,\sigma^2-\beta\right)}{(N+2\alpha-2)^2\,\sigma^5}$$

The only non-negative zero is at $\sigma^\star(\alpha,\beta) = \sqrt{\frac{\beta}{\alpha-1}}$ which is the minimizer of $T(\sigma)$. Note this corresponds to $\tau = \frac{\alpha-1}{\beta}$, the mode of the prior, which is as we expected. The minimum value of this ratio is then:

$$G(\alpha,N) \doteq T(\sigma^\star) = \frac{N^2}{(N+2\alpha-2)^2}$$

and this represents the largest realization-specific "gain" occurring at the prior mode, of the Bayesian estimator over an efficient unbiased estimator. Figure 5.7 shows $G(\alpha,N)$ for several values of $N$. We can also quantify estimator performance in low $f(\tau)$ regions by defining two measures:

$$\tilde{G}(\alpha,N) \doteq \lim_{\sigma\to\infty} T(\sigma) = \frac{\left[2\,(\alpha-1)^2+N\right]N}{(N+2\alpha-2)^2}$$
$$L(\alpha,\beta,N) \doteq \sigma^\star(\alpha,\beta) - \widehat{\sigma}(\alpha,\beta,N)$$

where $\widehat{\sigma}(\alpha,\beta,N)$ is the solution of $[T(\sigma)=1]$ over $(0,\sigma^\star)$.
$\tilde{G}(\alpha,N)$ represents the asymptotic "efficiency" of the Bayes estimator as $\sigma\to\infty$ while $L(\alpha,\beta,N)$ is the width of the interval to the left of $\sigma^\star$ where super-efficiency is achieved. The calculations for $L(\alpha,\beta,N)$ are lengthy and not shown here, it is plotted in figure 5.8 for several values of $N$. Also shown in figure 5.9 is $\tilde{G}(\alpha,N)$ which does not depend on $\beta$.

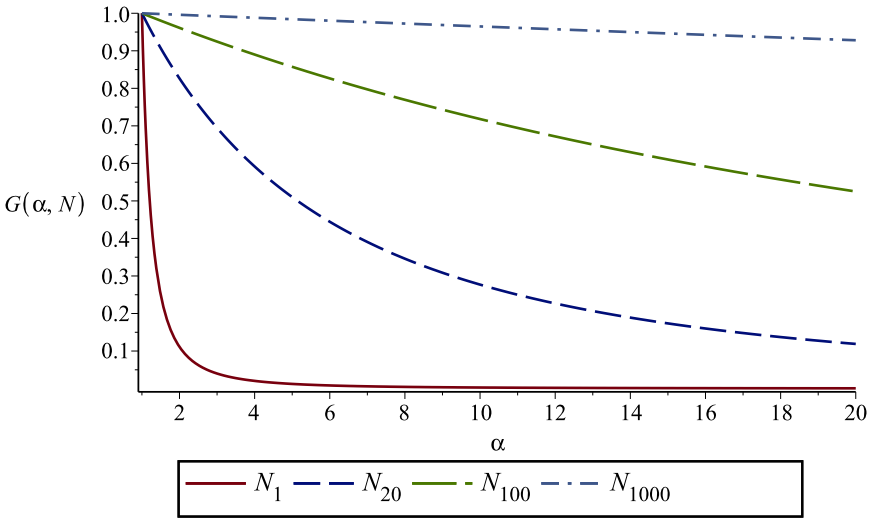


**Figure 5.7:** Minimum value of MSE/CRLB ratio

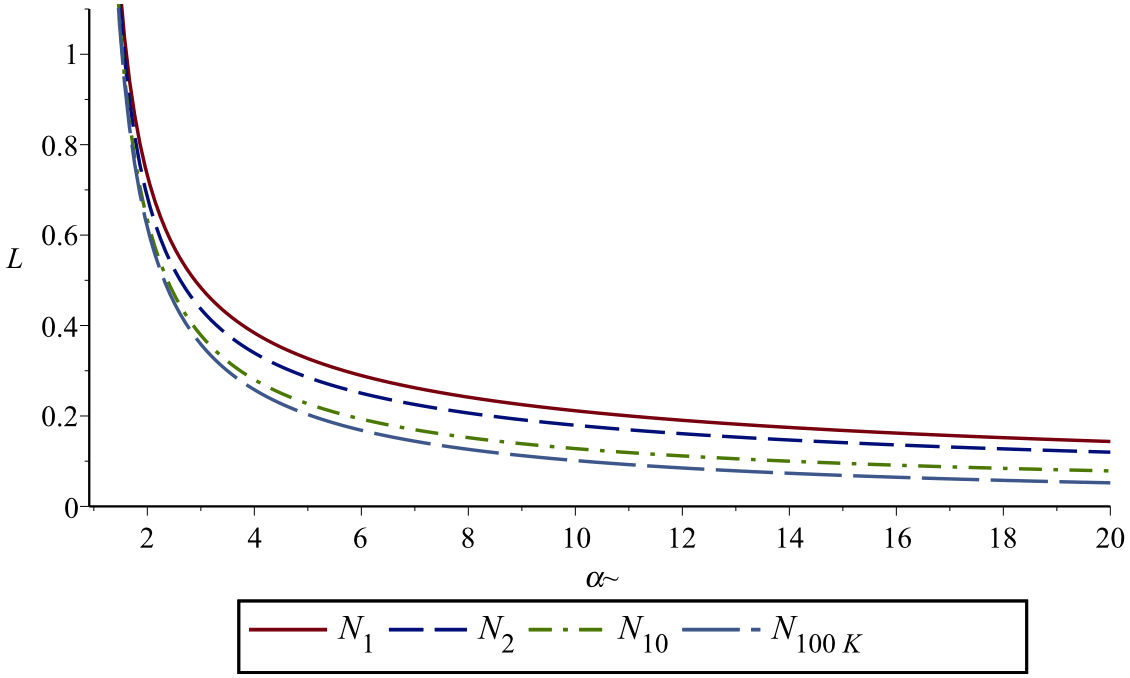


**Figure 5.8:** Width of super-efficiency interval for $\beta = 3$

We can make several observations concerning the influence of the hyperparameters and sample size on realization-specific MSE for this example. The width of super-efficiency decreases rapidly as the prior becomes more informative due to increasing $\alpha$ and is weakly affected by increasing N. The asymptotic efficiency $\tilde{G}(\alpha, N)$ is highly sensitive to both $\alpha$ and $N$. With very small data sets (single-digit) the Bayes estimator performs significantly better than the uniformly minimum variance unbiased estimator (UMVUE) even for large values of $\sigma$ (these are outliers with respect

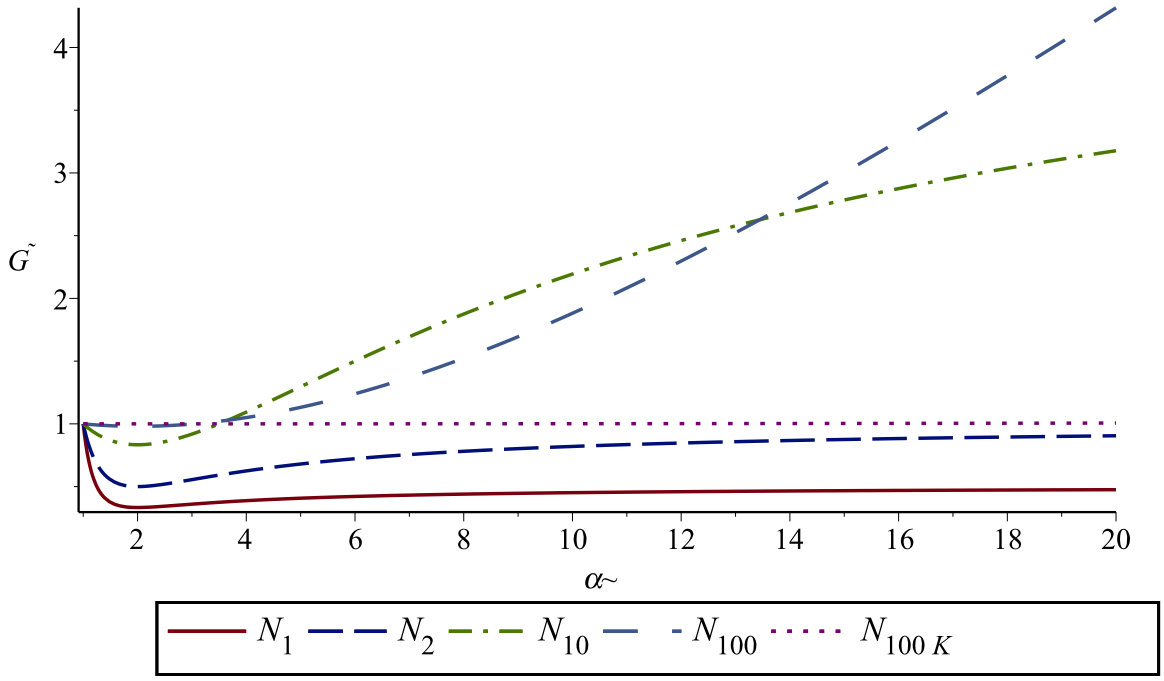


**Figure 5.9:** Efficiency of Bayes estimator as $\sigma \to \infty$

to the prior) and yields a large gain in efficiency close to the prior mode. However, as the sample set grows the situation reverses; efficiency gains around the prior mode decrease and inefficiency grows at outliers of the prior. This is inline with the observation made about (5.30); a "stronger" data set will "fight" our prior more and result in generally worse performance at outliers of $f(\tau)$.

One should be cautious to draw conclusions based on this analysis concerning the superiority of the Bayes estimator over a UMVUE even for this specific example. The two estimators are representative of parallel paradigms in estimation theory and thus any direct performance comparison is entirely subjective. This analysis is meant to serve more as a supplement in evaluating how a Bayes estimator will perform at specific parameter realizations, and how this performance is influenced by variables in the problem (form of prior, sample size, etc.). It can be argued that many of the results presented herein are for a narrow scope of estimation problems. Unbiased estimators achieving efficiency under the CRB are relatively rare; it was proven in [77] that in a typical estimation problem with additive noise, the signal must be affine in the parameters and the noise, Gaussian. While reparametrization of the problem can somewhat relax this restriction in the case of the BCRB, the set of efficient estimators under this bound is also relatively restrictive. If there does not exist an estimator which can achieve the BCRB, then the bias function of the MMSE estimator will have a different form than shown in this paper, thus invalidating the bounds placed on realization-specific MSE. However, the generalized results and the methodology employed can indeed be applied to successively tighter (and achievable in some restricted sense) Bayesian

bounds covering a wider range of distributions. Specific problems have been found for which tighter bounds are achievable, for example see [71]. It is expected that one may directly apply the theory developed here to these cases. In [72], a promising technique is offered that perhaps can be used towards specific application of the preceding development in a much less restricted set of problems. Furthermore, it can be said that even in the case of the CRB itself, the requirement of unbiasedness (or having a known bias) creates a limitation on its usefulness. This is not believed to be true; the CRB is frequently used by practitioners in problems where existence of an unbiased estimator has not been established. While this may not be entirely valid, it does allow one to establish a useful "starting point" for performance evaluation.

## 5.4 Extension to Sequential Estimation

### 5.4.1 Overview

The ultimate goal of the work completed in the preceding sections of this chapter is to establish a foundation for development of results which can be geared towards the application of focus in this thesis; namely target-tracking in a Bayesian sequential estimation (specifically discrete-time filtering) context. Overall motivation for this has been discussed previously and is reiterated here: synthesis of the ability to extract information regarding best-attainable performance of a tracking algorithm conditioned on a *specific trajectory realized by the target*.

A substantial amount of literature exists that establishes conventional classical/Bayesian bounds for sequential estimation and is thoroughly covered in [67]. Feasible computation of the BCRLB, which is called the Posterior-Cramer-Rao-Bound (PCRB) in this context, for discrete-time filtering, prediction, and smoothing was made possible by the results in [78],[79], and [80]. Note that the analysis which is conducted here does not exactly fit into the same category as the aforementioned bounds. The disparity is made clear by examining the differences in how simulations are conducted in each scenario. Consider experimental evaluation of the Bayesian MSE for some random parameter; the BMSE can be computed by averaging the error over a large number of trials, with each consisting of a specific value of the parameter that is drawn from the prior, and observations drawn from the likelihood function conditioned on the drawn parameter value. This is in contrast to the scenario under consideration, where one must compute the *realization-specific* MSE by averaging over trials each consisting of the same parameter value. This is more inline with a classical estimation scenario, however the distinction lies in allowing estimation to be conducted utilizing prior information.

A similar analysis to the approach that is followed here can be found in section 4.3.1 of [79] and originally considered in [81] to some extent. There, the concept of fictitious measurements is used to incorporate prior knowledge into a classical estimation scenario; bounds are established

based on the ability of an estimator to utilize this additional information in conjunction with the actual measurement set to yield improvements in performance. This approach does appear to illustrate the improvement in best possible estimator performance that is afforded by the additional prior information at given parameter realizations. However, it is not clear whether this method is capable of predicting how performance will vary for an *arbitary* parameter realization; i.e. it is an optimistic bound. An alternative approach is presented herein that is more naturally based on "optimal" estimator bias for a given prior and it is shown how this approach is capable of predicting this performance variation.

### 5.4.2 Notation and Preliminary Definitions

Two distinct notations are assumed for the results presented in the foregoing section. In presenting general vector parameter estimation results, a K-dimensional vector of random parameters is denoted by $\boldsymbol{\theta}=[\theta_1\ \theta_2\ ...\ \theta_K]^\top\in\mathbb{R}^K$, which is to be estimated by an N-dimensional vector of scalar observations $\mathbf{Y}=[y_1\ y_2\ ...\ y_N]^\top\in\mathbb{R}^N$. Later when we specialize to sequential estimation, the notation is modified to represent a $T$-duration sequence of parameter vectors (or target trajectory), denoted $\mathbf{X}_{1:T}=\left[X_1^\top X_2^\top ... X_T^\top\right]^\top$, where it is understood that the components of this trajectory vector are themselves K-dimensional vectors (i.e. $\mathbf{X}_{1:T}\in\mathbb{R}^K\times\mathbb{R}^T$). The trajectory is then estimated based on a $T$-sequence of Q-dimensional vector observations, denoted $\mathbf{Y}_{1:T}=\left[Y_1^\top Y_2^\top ... Y_T^\top\right]^\top\in\mathbb{R}^Q\times\mathbb{R}^T$. In the general case, an estimator is a mapping $\hat{\boldsymbol{\theta}}:\mathbb{R}^N\to\mathbb{R}^K$ from the observation space to the parameter space. It is assumed the parameter/observation vector pair has a differentiable joint probability density function $f_{\mathbf{Y},\boldsymbol{\theta}}(Y,\theta)$ with corresponding likelihood function $f_{\mathbf{Y}|\boldsymbol{\theta}}(Y|\theta)$ and prior density $f_{\boldsymbol{\theta}}(\theta)$. Expectation over the parameter/observation space is denoted as $\mathbb{E}_{\mathbf{Y},\boldsymbol{\theta}}$, expectation conditioned on a particular $\boldsymbol{\theta}$ as $\mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}$, and expectation over the prior only as $\mathbb{E}_{\boldsymbol{\theta}}$. The bias of an estimator is here viewed as a functional which maps an estimator to the parameter space and is defined as:

$$\mathbf{b}(\boldsymbol{\theta})=\mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}\left[(\hat{\boldsymbol{\theta}}(Y)-\boldsymbol{\theta})\right]=\int_{\mathbb{R}^N}(\hat{\boldsymbol{\theta}}(Y)-\boldsymbol{\theta})f_{\mathbf{Y}|\boldsymbol{\theta}}(Y|\boldsymbol{\theta})dY \tag{5.42}$$

The conditional (or realization-specific) MSE matrix of an estimator is a functional that maps an estimator to $\mathbb{R}^K\times\mathbb{R}^K$, defined as:

$$\begin{aligned}\mathbf{MSE}(\boldsymbol{\theta})=&\mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}\left[(\hat{\boldsymbol{\theta}}(\mathbf{Y})-\boldsymbol{\theta})(\hat{\boldsymbol{\theta}}(\mathbf{Y})-\boldsymbol{\theta})^\top\right]\\&=\int_{\mathbb{R}^N}(\hat{\boldsymbol{\theta}}(\mathbf{Y})-\boldsymbol{\theta})(\hat{\boldsymbol{\theta}}(\mathbf{Y})-\boldsymbol{\theta})^\top f_{\mathbf{Y}|\boldsymbol{\theta}}(Y|\boldsymbol{\theta})dY\end{aligned} \tag{5.43}$$

The vector "score" $\mathbf{S}_{(\cdot)}:\mathbb{R}^K\to\mathbb{R}^K$ is defined for all assumed probability densities as:

$$\left[\mathbf{S}_{(\cdot)}\right]_i=\frac{\partial\log f_{(\cdot)}(\cdot)}{\partial\theta_i}$$

For example, an element of the joint score $\mathbf{S}_{\mathbf{Y},\boldsymbol{\theta}}$ reads:

$$[\mathbf{S}_{\mathbf{Y},\boldsymbol{\theta}}]_i = \frac{\partial \log f_{\mathbf{Y},\boldsymbol{\theta}}(Y,\theta)}{\partial \theta_i}$$

It is assumed that regularity conditions are satisfied for all densities, i.e. :

$$\mathbb{E}_{\mathbf{Y},\boldsymbol{\theta}}[\mathbf{S}_{\mathbf{Y},\boldsymbol{\theta}}] = 0, \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}[\mathbf{S}_{\mathbf{Y}|\boldsymbol{\theta}}] = 0, \mathbb{E}_{\boldsymbol{\theta}}[\mathbf{S}_{\boldsymbol{\theta}}] = 0$$

The vector operator of first partial derivatives with respect to a $K \times 1$ vector $\boldsymbol{\theta}$ is also denoted as $\bigtriangledown_{\boldsymbol{\theta}} = \left[\frac{\partial}{\partial\theta_1} \ \frac{\partial}{\partial\theta_2} \ \cdots \ \frac{\partial}{\partial\theta_K}\right]^\top$

Information matrices that we will subsequently work with are denoted as:

$$\mathbf{J}_B = \mathbb{E}_{\mathbf{Y},\boldsymbol{\theta}}\left[-\bigtriangledown_{\boldsymbol{\theta}}\bigtriangledown_{\boldsymbol{\theta}}^\top \log f(\mathbf{Y},\boldsymbol{\theta})\right] \qquad \text{Bayesian Information}$$

$$\mathbf{J}_D(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}\left[-\bigtriangledown_{\boldsymbol{\theta}}\bigtriangledown_{\boldsymbol{\theta}}^\top \log f(\mathbf{Y}|\boldsymbol{\theta})\right] \qquad \text{Fisher Information}$$

$$\mathbf{J}_P = \mathbb{E}_{\boldsymbol{\theta}}\left[-\bigtriangledown_{\boldsymbol{\theta}}\bigtriangledown_{\boldsymbol{\theta}}^\top \log f(\boldsymbol{\theta})\right] \qquad \text{Prior Information}$$

The Bayesian Cramer Rao Bound (BCRLB) bounds the Bayesian MSE and is expressed as:

$$\mathbf{BMSE} = \mathbb{E}_{\mathbf{Y},\boldsymbol{\theta}}\left[\mathbf{MSE}(\boldsymbol{\theta})\right] \succeq \mathbf{J}_B^{-1} \tag{5.44}$$

where $A \succeq B$ is understood to mean that $A - B$ is positive semidefinite and it is assumed that $\mathbf{J}_B$ is nonsingular. The BCRLB is achievable if there exists a function (estimator) $\hat{\boldsymbol{\theta}}(\mathbf{Y})$ such that the following equality holds for some matrix $\mathbf{C}$ independent of $\mathbf{Y}$ and $\boldsymbol{\theta}$:

$$(\hat{\boldsymbol{\theta}}(Y) - \boldsymbol{\theta}) = \mathbf{C}\mathbf{S}_{\mathbf{Y},\boldsymbol{\theta}} \tag{5.45}$$

$\hat{\boldsymbol{\theta}}(\mathbf{X})$ is then the optimal Bayesian estimator and it can be shown that $\mathbf{C} = \mathbf{J}_B^{-1}$. Following the approach in [76] the following identity is derived:

$$\mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}\left[\mathbf{S}_{\mathbf{Y},\boldsymbol{\theta}}\mathbf{S}_{\mathbf{Y},\boldsymbol{\theta}}^\top\right] = \mathbf{J}_D(\boldsymbol{\theta}) + \mathbf{S}_{\boldsymbol{\theta}}\mathbf{S}_{\boldsymbol{\theta}}{}^\top \tag{5.46}$$

This can be developed by starting with the following trivial identity:

$$\int \frac{\partial}{\partial\theta_i} f(\mathbf{Y},\boldsymbol{\theta}) d\mathbf{Y} = \int \frac{\partial \log f(\mathbf{Y},\theta)}{\partial\theta_i} f(\mathbf{Y},\boldsymbol{\theta}) d\mathbf{Y} = \frac{\partial f(\boldsymbol{\theta})}{\partial\theta_i}$$

Differentiating by $\theta_j$ we then have:

$$\left(\int \frac{\partial \log f(\mathbf{Y},\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial \log f(\mathbf{Y}|\boldsymbol{\theta})}{\partial\theta_j} f(\mathbf{Y}|\boldsymbol{\theta}) d\mathbf{Y} + \int \frac{\partial^2 \log f(\mathbf{Y},\boldsymbol{\theta})}{\partial\theta_j\partial\theta_i} f(\mathbf{Y}|\boldsymbol{\theta}) d\mathbf{Y}\right) = \frac{\partial^2 \log f(\boldsymbol{\theta})}{\partial\theta_j\partial\theta_i}$$

The first term on the left-hand side is:

$$\int \frac{\partial \log f(\mathbf{Y},\boldsymbol{\theta})}{\partial \theta_i}\frac{\partial \log f(\mathbf{Y}|\boldsymbol{\theta})}{\partial \theta_j} f(\mathbf{Y}|\boldsymbol{\theta})d\mathbf{Y} =$$
$$\int \frac{\partial \log f(\mathbf{Y},\boldsymbol{\theta})}{\partial \theta_i}\frac{\partial \log f(\mathbf{Y},\boldsymbol{\theta})}{\partial \theta_j} f(\mathbf{Y}|\boldsymbol{\theta})d\mathbf{Y} - \frac{\partial \log f(\boldsymbol{\theta})}{\partial \theta_j}\int \frac{\partial \log f(\mathbf{Y},\boldsymbol{\theta})}{\partial \theta_i} f(\mathbf{Y}|\boldsymbol{\theta}d\mathbf{Y}$$

and the second term is:

$$\int \frac{\partial^2 \log f(\mathbf{Y},\boldsymbol{\theta})}{\partial \theta_j \partial \theta_i} f(\mathbf{Y}|\boldsymbol{\theta})d\mathbf{Y} = \int \frac{\partial^2 \log f(\mathbf{Y}|\boldsymbol{\theta})}{\partial \theta_j \partial \theta_i} f(\mathbf{Y}|\boldsymbol{\theta})d\mathbf{Y} + \frac{\partial^2 \log f(\boldsymbol{\theta})}{\partial \theta_j \partial \theta_i}$$

Simplifying the full expression we have:

$$\left( \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}\left[\frac{\partial^2 \log f(\mathbf{Y}|\boldsymbol{\theta})}{\partial \theta_j \partial \theta_i}\right] + \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}\left[\frac{\partial \log f(\mathbf{Y},\boldsymbol{\theta})}{\partial \theta_i}\frac{\partial \log f(\mathbf{Y},\boldsymbol{\theta})}{\partial \theta_j}\right] - \frac{\partial \log f(\boldsymbol{\theta})}{\partial \theta_j}\frac{\partial \log f(\boldsymbol{\theta})}{\partial \theta_i} \right) = 0$$

which is equivalent to the identity in (5.46).

### 5.4.3 Vector Parameter Estimation

Similar to the case for a scalar parameter, a straightforward extension can be made to a vector parameter. Specifically, any efficient Bayesian estimator of the $K$-dimensional parameter vector $\boldsymbol{\theta}$ which achieves the BCRLB is equivalent to a classical biased estimator with bias function $\mathbf{b}^\star(\boldsymbol{\theta}) = \mathbf{J}_B^{-1}\mathbf{S}_{\boldsymbol{\theta}}$. This can easily be shown by first noting that by assumption, $\hat{\boldsymbol{\theta}}(Y)$ is efficient and satisfies (5.45), as a result:

$$\mathbf{b}^\star(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}\left[(\hat{\boldsymbol{\theta}}(Y) - \boldsymbol{\theta})\right] = \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}\left[\mathbf{C}\mathbf{S}_{\mathbf{Y},\boldsymbol{\theta}}\right] = \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}\left[\mathbf{C}\mathbf{S}_{\mathbf{Y}|\boldsymbol{\theta}}\right] + \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}\left[\mathbf{C}\mathbf{S}_{\boldsymbol{\theta}}\right] = \mathbf{C}\mathbf{S}_{\boldsymbol{\theta}}$$

This leads to an important result concerning realization-specific performance; namely that any efficient Bayesian estimator of the $K$-dimensional parameter vector $\boldsymbol{\theta}$ has MSE matrix at a given parameter realization:

$$\mathbf{MSE}(\boldsymbol{\theta}) = \mathbf{J}_\mathbf{B}^{-1}\left(\mathbf{J}_\mathbf{D}(\boldsymbol{\theta}) + \mathbf{S}_{\boldsymbol{\theta}}\mathbf{S}_{\boldsymbol{\theta}}{}^\top\right)\mathbf{J}_\mathbf{B}^{-1} \tag{5.47}$$

One can easily see this by noting that due to (5.45) , $(\hat{\boldsymbol{\theta}}(Y) - \boldsymbol{\theta}) = \mathbf{J}_\mathbf{B}^{-1}\mathbf{S}_{\mathbf{Y},\boldsymbol{\theta}}$ so we have:

$$\mathbf{MSE}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}\left[(\hat{\boldsymbol{\theta}}(Y) - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}}(Y) - \boldsymbol{\theta})^\top\right] = \mathbf{J}_\mathbf{B}^{-1}\mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}}\left(\mathbf{S}_{\mathbf{Y},\boldsymbol{\theta}}\mathbf{S}_{\mathbf{Y},\boldsymbol{\theta}}^\top\right)\mathbf{J}_\mathbf{B}^{-1} \tag{5.48}$$

Substituting (5.46) into (5.48) confirms (5.47). From (5.47) we can see that this can also be written as:

$$\mathbf{MSE}(\boldsymbol{\theta}) = \mathbf{J}_\mathbf{B}^{-1}\mathbf{J}_\mathbf{D}(\boldsymbol{\theta})\mathbf{J}_\mathbf{B}^{-1} + \mathbf{b}^\star(\boldsymbol{\theta})\mathbf{b}^\star(\boldsymbol{\theta})^\top \tag{5.49}$$

### 5.4.4 Sequential Estimation

The preceding results are now applied to the case of efficient discrete-time filtering. Let us consider estimation of the $(K \times 1)$ parameter vector at time $T$ denoted $\mathbf{X}_T$ based on the sequence of $(Q \times 1)$ vector observations occurring from $t = 1$ to $T$; $\mathbf{Y}_{1:T} = \left[\mathbf{Y}_1^\top \ \mathbf{Y}_2^\top \ ... \mathbf{Y}_T^\top\right]^\top$. A trajectory of the state is also defined as $\mathbf{X}_{1:T} = \left[\mathbf{X}_1^\top \ \mathbf{X}_2^\top \ ... \mathbf{X}_T^\top\right]^\top$, which represents a realization of all states up to time $T$. An estimator of $\mathbf{X}_T$ based on $\mathbf{Y}_{1:T}$ is the mapping $\hat{\mathbf{X}}_T : \mathbb{R}^{TQ} \to \mathbb{R}^K$ that maps the trajectory of observation vectors from time 1 to $T$ to a vector in the parameter space at time $T$. The Bayesian MSE matrix of $\hat{\mathbf{X}}_T$ depends on the joint statistics of the observation and parameter trajectories, $f(\mathbf{Y}_{1:T}, \mathbf{X}_{1:T})$, and can be bounded with the BCRLB by considering the $(KT \times KT)$ Bayesian MSE matrix of the full parameter trajectory up to time $T$ as follows:

$$\mathbf{BMSE}_{1:T} \succeq \mathbf{J}_{\mathbf{B},1:T}^{-1} \tag{5.50}$$

where $\mathbf{J}_{(\cdot),1:T}$ denotes the corresponding trajectory information matrix at time $T$, i.e. $\mathbf{J}_{\mathbf{D},1:T}(\mathbf{X}_{1:T}) = \mathbb{E}_{\mathbf{Y}_{1:T}|\mathbf{X}_{1:T}}\left[\mathbf{S}_{\mathbf{Y}_{1:T}|\mathbf{X}_{1:T}}\mathbf{S}_{\mathbf{Y}_{1:T}|\mathbf{X}_{1:T}}^\top\right]$. Partitioning $\mathbf{J}_{\mathbf{B},1:T}$ into $(T \times T)$ blocks each with dimension $(K \times K)$, it should be clear that the BMSE matrix of $\hat{\boldsymbol{\theta}}_T$ alone is bounded as:

$$\mathbf{BMSE}_T \succeq \mathbf{J}_{\mathbf{B},T}^{-1} \tag{5.51}$$

where $\mathbf{J}_{\mathbf{B},T}^{-1}$ or $\left[\mathbf{J}_{\mathbf{B},1:T}^{-1}\right]_{T,T}$ denotes the $(T,T)$-th block of $\mathbf{J}_{\mathbf{B},1:T}^{-1}$. It is worth pointing out with this notation that $\mathbf{J}_{\mathbf{B},T}^{-1} \neq [\mathbf{J}_{\mathbf{B},1:T}]_{T,T}^{-1}$. The optimal bias function at time $T$ is a $(K \times 1)$ vector which depends on the specific state trajectory from time 1 to $T$ and is written as $\mathbf{b}_T(\mathbf{X}_{1:T})$. Its value can be determined by considering the bias of the full trajectory, which is a $(TK \times 1)$ vector:

$$\mathbf{b}_{1:T}^\star(\mathbf{X}_{1:T}) = \mathbf{J}_{\mathbf{B},1:T}^{-1}\mathbf{S}_{\mathbf{X}_{1:T}} \tag{5.52}$$

It can then be stated that any efficient Bayesian estimator for the discrete-time filtering problem conditioned on a particular trajectory, $\mathbf{X}_{1:T}$, has MSE matrix at time $T$ equal to:

$$\mathbf{MSE}_T(\mathbf{X}_{1:T}) = \left[\ \mathbf{J}_{\mathbf{B},1:T}^{-1}\mathbf{J}_{\mathbf{D},1:T}(\mathbf{X}_{1:T})\mathbf{J}_{\mathbf{B},1:T}^{-1} + \mathbf{b}_{1:T}^\star(\mathbf{X}_{1:T})\mathbf{b}_{1:T}^\star(\mathbf{X}_{1:T})^\top\ \right]_{T,T} \tag{5.53}$$

This can be seen as a trivial extension of the results in section 5.4.3 by noting that estimation of $\mathbf{X}_T$ in the filtering problem is equivalent to estimation of the $T$-th segment of $K$ components within the parameter vector $\mathbf{X}_{1:T}$. The main difference here being that the full observation vector $\mathbf{Y}_{1:T}$ consists of $T$ segments of $(Q \times 1)$ vectors, whereas estimation in section 5.4.3 was based on the single-segment vector $\mathbf{Y}$ of $N$ components.

Direct computation of (5.53) requires inversion of the full Bayesian trajectory information matrix which grows in dimension with increasing $T$. The problem is greatly simplified for Markovian processes with conditionally independent measurements, and in this case, a recursive form for computation of $\mathbf{J}_{\mathbf{B},T}^{-1}$ (also known as the PCRB) was developed in [78]. It will be shown how a recursive

form can also be developed for (5.53). As there is a reliance on many of the same terms as in [78] and [82] to express this form, some of the derivation will be identical. For brevity, details are omitted whenever deemed unnecessary.

Let us assume the state evolution is Markovian and the observation vector at time $T$, $\mathbf{Y}_T$, is dependent only on $\mathbf{X}_T$. One can then express the joint density of the observation and parameter trajectories as:

$$f(\mathbf{Y}_{1:T}, \mathbf{X}_{1:T}) = f(\mathbf{X}_0)\prod_{t=1}^{T} f(\mathbf{X}_t|\mathbf{X}_{t-1})\prod_{t=1}^{T} f(\mathbf{Y}_t|\mathbf{X}_t) \tag{5.54}$$

The trajectory information matrix can be partitioned as:

$$\mathbf{J}_{\mathbf{B},1:T} = \begin{bmatrix} \mathbf{A}_{1:T} & \mathbf{B}_{1:T} \\ \mathbf{B}_{1:T}^{\top} & \mathbf{D}_{1:T} \end{bmatrix} \tag{5.55}$$

where $\mathbf{A}_{1:T}$ is $((T-1)K \times (T-1)K)$, $\mathbf{B}_{1:T}$ is $((T-1)K \times K)$ and $\mathbf{D}_{1:T}$ is $(K \times K)$. Using block inversion we can express $\mathbf{J}_{\mathbf{B},1:T}^{-1}$ as :

$$\mathbf{J}_{\mathbf{B},1:T}^{-1} = \begin{bmatrix} \left[\mathbf{J}_{\mathbf{B},1:T}^{-1}\right]_{1,1} & \left[\mathbf{J}_{\mathbf{B},1:T}^{-1}\right]_{1,2} \\ \left[\mathbf{J}_{\mathbf{B},1:T}^{-1}\right]_{1,2}^{T} & \mathbf{J}_{\mathbf{B},T}^{-1} \end{bmatrix} \tag{5.56}$$

with,

$$\begin{aligned} \left[\mathbf{J}_{\mathbf{B},1:T}^{-1}\right]_{1,1} &= \left(\mathbf{A}_{1:T} - \mathbf{B}_{1:T}\mathbf{D}_{1:T}^{-1}\mathbf{B}_{1:T}^{\top}\right)^{-1} \\ \left[\mathbf{J}_{\mathbf{B},1:T}^{-1}\right]_{1,2} &= -\mathbf{A}_{1:T}^{-1}\mathbf{B}_{1:T}\left(\mathbf{D}_{1:T} - \mathbf{B}_{1:T}\mathbf{A}_{1:T}^{-1}\mathbf{B}_{1:T}^{\top}\right)^{-1} \\ \mathbf{J}_{\mathbf{B},T}^{-1} &= \left(\mathbf{D}_{1:T} - \mathbf{B}_{1:T}\mathbf{A}_{1:T}^{-1}\mathbf{B}_{1:T}^{\top}\right)^{-1} \end{aligned} \tag{5.57}$$

Due to the Markovian assumption, $\mathbf{J}_{\mathbf{B},1:T}$ has the form:

$$\mathbf{J}_{\mathbf{B},1:T} = \begin{bmatrix} \mathbf{J}_{B,0}(\mathbf{X_0}) & \mathbf{V}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{V}_1^{\top} & \mathbf{W}_1 + \mathbf{U}_2 & \mathbf{V}_2 & \ddots & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & \ddots & \mathbf{V}_{T-1}^{\top} & \mathbf{W}_{T-1} + \mathbf{U}_T & \mathbf{V}_T \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{V}_T^{\top} & \mathbf{W}_T \end{bmatrix} \tag{5.58}$$

where,

$$\mathbf{V}_T = \mathbb{E}_{\mathbf{Y},\boldsymbol{\theta}}\left[-\bigtriangledown_{\mathbf{X}_{T-1}}\bigtriangledown_{\mathbf{X}_T}^{\top}\log f(\mathbf{X}_T|\mathbf{X}_{T-1})\right] \tag{5.59}$$

$$\mathbf{U}_T = \mathbb{E}_{\mathbf{Y},\boldsymbol{\theta}}\left[-\bigtriangledown_{\mathbf{X}_{T-1}}\bigtriangledown_{\mathbf{X}_{T-1}}^{\top}\log f(\mathbf{X}_T|\mathbf{X}_{T-1})\right] \tag{5.60}$$

$$\mathbf{W}_T = \mathbb{E}_{\mathbf{Y},\boldsymbol{\theta}}\left[-\bigtriangledown_{\mathbf{X}_T}\bigtriangledown_{\mathbf{X}_T}^{\top}\log f(\mathbf{X}_T|\mathbf{X}_{T-1})\right] + \mathbb{E}_{\mathbf{Y},\boldsymbol{\theta}}\left[-\bigtriangledown_{\mathbf{X}_T}\bigtriangledown_{\mathbf{X}_T}^{\top}\log f(\mathbf{Y}_T|\mathbf{X}_T)\right] \tag{5.61}$$

So $\mathbf{J}_{\mathbf{B},1:T}$ can be further partitioned as:

$$\mathbf{J}_{\mathbf{B},1:T} = \begin{bmatrix} \begin{bmatrix} \mathbf{A}_{1:T-1} & \mathbf{B}_{1:T-1} \\ \mathbf{B}_{1:T-1}^\top & \mathbf{D}_{1:T-1}+\mathbf{U}_T \end{bmatrix} & \mathbf{B}_{1:T} \\ \mathbf{B}_{1:T}^\top & \mathbf{D}_{1:T} \end{bmatrix} \tag{5.62}$$

where $\mathbf{D}_{1:T} = \mathbf{W}_T$ and $\mathbf{B}_{1:T} = \begin{bmatrix} 0 & \mathbf{V}_T^\top \end{bmatrix}^\top$.

Referring back to (5.57), we have:

$$\begin{aligned} \mathbf{J}_{\mathbf{B},T}^{-1} &= \left( \mathbf{D}_{1:T} - \mathbf{B}_{1:T} \mathbf{A}_{1:T}^{-1} \mathbf{B}_{1:T}^\top \right)^{-1} \\ &= \left( \mathbf{W}_T - \begin{bmatrix} \mathbf{0} & \mathbf{V}_T^\top \end{bmatrix} \mathbf{A}_{1:T}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{V}_T \end{bmatrix} \right)^{-1} \\ &= \left( \mathbf{W}_T - \mathbf{V}_T^\top \left[ \mathbf{A}_{1:T}^{-1} \right]_{2,2} \mathbf{V}_T \right)^{-1} \end{aligned} \tag{5.63}$$

with $\left[ \mathbf{A}_{1:T}^{-1} \right]_{2,2}$ as the lower-right block of the inverse of the partitioned $\mathbf{A}_{1:T}$ shown as the upper left matrix in (5.62). Evaluating block inversion on this partition yields:

$$\begin{aligned} \left[ \mathbf{A}_{1:T}^{-1} \right]_{2,2} &= \left( (\mathbf{D}_{1:T-1} + \mathbf{U}_T) - \mathbf{B}_{1:T-1}^\top \mathbf{A}_{1:T-1}^{-1} \mathbf{B}_{1:T-1} \right)^{-1} \\ &= \left( \mathbf{U}_T + \mathbf{J}_{\mathbf{B},T-1} \right)^{-1} \end{aligned} \tag{5.64}$$

Yielding the recursive form for $\mathbf{J}_{\mathbf{B},T}^{-1}$:

$$\mathbf{J}_{\mathbf{B},T}^{-1} = \left( \mathbf{W}_T - \mathbf{V}_T^\top \left( \mathbf{U}_T + \mathbf{J}_{\mathbf{B},T-1} \right)^{-1} \mathbf{V}_T \right)^{-1} \tag{5.65}$$

For what follows, we will also need to develop a recursive form for the $((T-1)K \times K)$ upper right block $\left[ \mathbf{J}_{\mathbf{B},1:T}^{-1} \right]_{1,2}$ of (5.56). Referring again to the partition of $\mathbf{A}_{1:T}$ in (5.62) we can write:

$$\left[ \mathbf{J}_{\mathbf{B},1:T}^{-1} \right]_{1,2} = -\mathbf{A}_{1:T} \mathbf{B}_{1:T} \mathbf{J}_{\mathbf{B},T}^{-1} = -\begin{bmatrix} \left[ \mathbf{A}_{1:T}^{-1} \right]_{1,2} \\ \left[ \mathbf{A}_{1:T}^{-1} \right]_{2,2} \end{bmatrix} \mathbf{V}_T \mathbf{J}_{\mathbf{B},T}^{-1} \tag{5.66}$$

$$\left[ \mathbf{A}_{1:T}^{-1} \right]_{1,2} = -\mathbf{A}_{1:T-1} \mathbf{B}_{1:T-1} \left( \mathbf{U}_T + \mathbf{J}_{\mathbf{B},T-1} \right)^{-1} \tag{5.67}$$

But it should be clear that:

$$\begin{aligned} &\left[ \mathbf{J}_{B,1:T-1}^{-1} \right]_{1,2} \mathbf{J}_{\mathbf{B},T-1} \left( \mathbf{U}_T + \mathbf{J}_{\mathbf{B},T-1} \right)^{-1} \\ &= -\mathbf{A}_{1:T-1}^{-1} \mathbf{B}_{1:T-1} \left( \mathbf{U}_T + \mathbf{J}_{\mathbf{B},T-1} \right)^{-1} = \left[ \mathbf{A}_{1:T}^{-1} \right]_{1,2} \end{aligned} \tag{5.68}$$

Resulting in:

$$\left[ \mathbf{J}_{B,1:T}^{-1} \right]_{1,2} = -\begin{bmatrix} \left[ \mathbf{J}_{B,1:T-1}^{-1} \right]_{1,2} \mathbf{J}_{\mathbf{B},T-1} \\ \mathbf{I}_{K \times K} \end{bmatrix} \left( \mathbf{U}_T + \mathbf{J}_{\mathbf{B},T-1} \right)^{-1} \mathbf{V}_T \mathbf{J}_{\mathbf{B},T}^{-1} \tag{5.69}$$

which is in fact the recursion:

$$\left[\mathbf{J}_{B,1:T}^{-1}\right]_{1,2} = \left[\mathbf{J}_{B,1:T-1}^{-1}\right]_{1:T-1,T-1} \mathbf{G}_T \tag{5.70}$$

where $\left[\mathbf{J}_{B,1:T-1}^{-1}\right]_{1:T-1,T-1}$ is defined as the $(T-1)$-th right column block of $\mathbf{J}_{B,1:T-1}^{-1}$ with dimension $((T-1)K \times K)$ and $\mathbf{G}_T$ is the $(K \times K)$ matrix:

$$\mathbf{G}_T = -\mathbf{J}_{\mathbf{B},T-1} \left(\mathbf{U}_T + \mathbf{J}_{\mathbf{B},T-1}\right)^{-1} \mathbf{V}_T \mathbf{J}_{\mathbf{B},T}^{-1} \tag{5.71}$$

Also, by noting that the lower $(K \times K)$ block of $\left[\mathbf{J}_{B,1:T}^{-1}\right]_{1,2}$ is $\left[\mathbf{J}_{\mathbf{B},1:T}^{-1}\right]_{T-1,T}$ we have:

$$\left[\mathbf{J}_{\mathbf{B},1:T}^{-1}\right]_{T-1,T} = -\left(\mathbf{U}_T + \mathbf{J}_{\mathbf{B},T-1}\right)^{-1} \mathbf{V}_T \mathbf{J}_{\mathbf{B},T}^{-1} \tag{5.72}$$

Now shifting attention to the prior score vector $\mathbf{S}_{\mathbf{X}_{1:T}}$, this term can be decomposed as:

$$\mathbf{S}_{\mathbf{X}_{1:T}} = \hat{\mathbf{S}}_{\mathbf{X}_{1:T-1}} + \mathbf{Z}_T \tag{5.73}$$

with:

$$\hat{\mathbf{S}}_{\mathbf{X}_{1:T-1}} = \begin{bmatrix} \bigtriangledown_{\mathbf{X}_0} \log f(\mathbf{X}_0) + \bigtriangledown_{\mathbf{X}_0} \log f(\mathbf{X}_1|\mathbf{X}_0) \\ \bigtriangledown_{\mathbf{X}_1} \log f(\mathbf{X}_1|\mathbf{X}_0) + \bigtriangledown_{\mathbf{X}_1} \log f(\mathbf{X}_2|\mathbf{X}_1) \\ \vdots \\ \bigtriangledown_{\mathbf{X}_{T-1}} \log f(\mathbf{X}_{T-1}|\mathbf{X}_{T-2}) \\ \mathbf{0} \end{bmatrix} \tag{5.74}$$

$$\mathbf{Z}_T = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \bigtriangledown_{\mathbf{X}_{T-1}} \log f(\mathbf{X}_T|\mathbf{X}_{T-1}) \\ \bigtriangledown_{\mathbf{X}_T} \log f(\mathbf{X}_T|\mathbf{X}_{T-1})) \end{bmatrix} \tag{5.75}$$

So we can split (5.52) into two terms as:

$$\mathbf{b}_{1:T}^{\star}(\mathbf{X}_{1:T}) = \mathbf{J}_{\mathbf{B},1:T}^{-1} \hat{\mathbf{S}}_{\mathbf{X}_{1:T-1}} + \mathbf{J}_{\mathbf{B},1:T}^{-1} \mathbf{Z}_T \tag{5.76}$$

Referring back to (5.53) we wish to determine the lower right $K \times K$ block:

$$\begin{aligned} \left[\mathbf{b}_{1:T}^{\star}(\mathbf{X}_{1:T}) \mathbf{b}_{1:T}^{\star}(\mathbf{X}_{1:T})^T\right]_{T,T} &= \left[\mathbf{b}_{1:T}^{\star}(\mathbf{X}_{1:T})\right]_T \left[\mathbf{b}_{1:T}^{\star}(\mathbf{X}_{1:T})\right]_T^{\top} \\ &= \mathbf{b}_T^{\star}(\mathbf{X}_{1:T}) \mathbf{b}_T^{\star}(\mathbf{X}_{1:T})^{\top} \end{aligned} \tag{5.77}$$

First we examine the $T$-th $(K \times 1)$ element of the $(TK \times 1)$ column vector $\mathbf{J}_{\mathbf{B},1:T}^{-1} \hat{\mathbf{S}}_{\mathbf{X}_{1:T-1}}$:

$$\begin{aligned} \left[\mathbf{J}_{\mathbf{B},1:T}^{-1} \hat{\mathbf{S}}_{\mathbf{X}_{1:T-1}}\right]_T &= \left(\sum_{i=1}^{\top} \left[\mathbf{J}_{\mathbf{B},1:T}^{-1}\right]_{T,i} \left[\hat{\mathbf{S}}_{\mathbf{X}_{1:T-1}}\right]_i\right) \\ &= \left[\mathbf{J}_{B,1:T}^{-1}\right]_{1,2}^{\top} \mathbf{S}_{\mathbf{X}_{1:T-1}} = \mathbf{G}_T^{\top} \left[\mathbf{J}_{B,1:T-1}^{-1}\right]_{T-1,1:T-1} \mathbf{S}_{\mathbf{X}_{1:T-1}} \end{aligned} \tag{5.78}$$

since the $T$-th $(K \times 1)$ element of $\hat{\mathbf{S}}_{\mathbf{X}_{1:T-1}}$ is $\mathbf{0}$ and the partition of $\mathbf{J}^{-1}_{\mathbf{B},1:T}$ for any $T$ is symmetric, i.e.

$$\left[\mathbf{J}^{-1}_{B,1:T-1}\right]^{\top}_{1:T-1,T-1} = \left[\mathbf{J}^{-1}_{B,1:T-1}\right]_{T-1,1:T-1}$$

We can also write:

$$\begin{aligned}\mathbf{b}^{\star}_{T}(\mathbf{X}_{1:T}) = [\mathbf{b}^{\star}_{1:T}(\mathbf{X}_{1:T})]_T &= \left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\mathbf{S}_{\mathbf{X}_{1:T}}\right]_T = \left[\mathbf{J}^{-1}_{B,1:T}\right]_{T,1:T}\mathbf{S}_{\mathbf{X}_{1:T}} \\ &= \left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\hat{\mathbf{S}}_{\mathbf{X}_{1:T-1}}\right]_T + \left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\mathbf{Z}_T\right]_T \end{aligned} \tag{5.79}$$

and obviously,

$$\mathbf{b}^{\star}_{T-1}(\mathbf{X}_{1:T-1}) = \left[\mathbf{b}^{\star}_{1:T-1}(\mathbf{X}_{1:T-1})\right]_{T-1} = \left[\mathbf{J}^{-1}_{B,1:T-1}\right]_{T-1,1:T-1}\mathbf{S}_{\mathbf{X}_{1:T-1}} \tag{5.80}$$

Noting that the first $T-2$ $(K \times 1)$ elements of $\mathbf{Z}_T$ are zero,

$$\widehat{\mathbf{b}}_T \triangleq \left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\mathbf{Z}_T\right]_T = \left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\right]_{T,T-1}[\mathbf{Z}_T]_{T-1} + \left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\right]_{T,T}[\mathbf{Z}_T]_T \tag{5.81}$$

Thus we have the recursive form for the optimal bias function:

$$\mathbf{b}^{\star}_{T}(\mathbf{X}_{1:T}) = \mathbf{G}^{\top}_{T}\mathbf{b}^{\star}_{T-1}(\mathbf{X}_{1:T-1}) + \widehat{\mathbf{b}}_T \tag{5.82}$$

The first term of (5.53) can be simplified by noting that due to the assumption of conditionally independent observations (i.e.$\mathbf{Y}_T$ is dependent only on $\mathbf{X}_T$ ), $\mathbf{J}_{\mathbf{D},1:T}(\mathbf{X}_{1:T})$ must be block-diagonal. Note the arguments of $\mathbf{J}_{\mathbf{D},1:T}(\mathbf{X}_{1:T})$ and $\mathbf{J}_{\mathbf{D},T}(\mathbf{X}_T)$ are made implicit in the subsequent derivation for space reasons, their actual value should be clear from context. One can then form the partition:

$$\mathbf{J}_{\mathbf{D},1:T} = \begin{bmatrix}\mathbf{J}_{\mathbf{D},1:T-1} & \mathbf{0}_{(T-1)K\times K} \\ \mathbf{0}_{K\times(T-1)K} & \mathbf{J}_{\mathbf{D},T}\end{bmatrix} \tag{5.83}$$

Utilizing block-multiplication with the partition of $\mathbf{J}^{-1}_{\mathbf{B},1:T}$ defined in (5.56) we arrive at the expression:

$$\mathbf{J}^{-1}_{\mathbf{B},1:T}\mathbf{J}_{\mathbf{D},1:T} = \begin{bmatrix}\left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\right]_{1,1}\mathbf{J}_{\mathbf{D},1:T-1} & \left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\right]_{1,2}\mathbf{J}_{\mathbf{D},T} \\ \left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\right]_{2,1}\mathbf{J}_{\mathbf{D},1:T-1} & \mathbf{J}^{-1}_{\mathbf{B},T}\mathbf{J}_{\mathbf{D},T}\end{bmatrix} \tag{5.84}$$

Postmultiplying this by $\mathbf{J}^{-1}_{\mathbf{B},1:T}$ and taking the lower right block (which is the $(T,T)$-th element) gives us:

$$\left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\mathbf{J}_{\mathbf{D},1:T}\mathbf{J}^{-1}_{\mathbf{B},1:T}\right]_{T,T} = \left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\right]_{2,1}\mathbf{J}_{\mathbf{D},1:T-1}\left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\right]_{1,2} + \mathbf{J}^{-1}_{\mathbf{B},T}\mathbf{J}_{\mathbf{D},T}\mathbf{J}^{-1}_{\mathbf{B},T} \tag{5.85}$$

Examining the first term on the RHS of (5.85):

$$\begin{aligned}&\left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\right]_{2,1}\mathbf{J}_{\mathbf{D},1:T-1}\left[\mathbf{J}^{-1}_{\mathbf{B},1:T}\right]_{1,2} = \\ &\mathbf{G}^{\top}_{T}\left[\mathbf{J}^{-1}_{B,1:T-1}\right]_{T-1,1:T-1}\mathbf{J}_{\mathbf{D},1:T-1}\left[\mathbf{J}^{-1}_{B,1:T-1}\right]_{1:T-1,T-1}\mathbf{G}_T\end{aligned} \tag{5.86}$$

*Complete the following steps at* $t = 0$:

*S1*: Set $\mathbf{J}_{B,\mathbf{0}} = \mathbf{J}_0(\mathbf{X_0})$

*S2*: Set $\mathbf{b}_{\mathbf{0}}^{\star}(\mathbf{X}_0) = \mathbf{J}_0(\mathbf{X_0})\left(\bigtriangledown_{\mathbf{X}_0} \log p(\mathbf{X_0})\right)$

*S3*: Set $\mathbf{M}_{q,\mathbf{0}} = \mathbf{0}_{K\times K}$

*Complete the following steps for* $t > 0$:

*S4*: Determine the $(K \times K)$ matrices $\mathbf{V}_T$, $\mathbf{U}_T$, and $\mathbf{W}_T$ using (5.59)-(5.61).

*S5*: Compute $\mathbf{J}_{\mathbf{B},T}^{-1}$ using (5.65).

*S6*: Compute $\left[\mathbf{J}_{\mathbf{B},1:T}^{-1}\right]_{T-1,T}$ using (5.72).

*S7*: Determine $\widehat{\mathbf{b}}_T$ using (5.81) with terms from the previous steps and $\mathbf{Z}_T$ from (5.75).

*S8*: Compute $\mathbf{b}_T^{\star}(\mathbf{X}_{1:T})$ using the recursion (5.82).

*S9*: Determine the $(K \times K)$ matrix: $\mathbf{M}_{d,T} = \mathbf{J}_{\mathbf{B},T}^{-1}\mathbf{J}_{\mathbf{D},T}(\mathbf{X}_T)\mathbf{J}_{\mathbf{B},T}^{-1}$.

*S10*: Compute $\mathbf{M}_{q,T}$ using the recursion (5.90).

*S11*: Finally compute $\mathbf{MSE}_T(\mathbf{X}_{1:T})$ using (5.91) and the preceding terms.

**Table 5.1:** Recursive Algorithm for Realization-Specific MSE Computation at time $T$.

But it is easy to show that:

$$\left[\mathbf{J}_{B,1:T-1}^{-1}\right]_{T-1,1:T-1}\mathbf{J}_{\mathbf{D},1:T-1}\left[\mathbf{J}_{B,1:T-1}^{-1}\right]_{1:T-1,T-1} = \tag{5.87}$$

$$\left[\mathbf{J}_{\mathbf{B},1:T-1}^{-1}\mathbf{J}_{\mathbf{D},1:T-1}\mathbf{J}_{\mathbf{B},1:T-1}^{-1}\right]_{T-1,T-1} \triangleq \mathbf{M}_{q,T-1} \tag{5.88}$$

Defining the second term in the RHS of (5.85) as $\mathbf{M}_{d,T}$ we then have the recursion:

$$\left[\mathbf{J}_{\mathbf{B},1:T}^{-1}\mathbf{J}_{\mathbf{D},1:T}\mathbf{J}_{\mathbf{B},1:T}^{-1}\right]_{T,T} = \mathbf{M}_{q,T} \tag{5.89}$$

where:

$$\mathbf{M}_{q,T} = \mathbf{G}_T^{\top}\mathbf{M}_{q,T-1}\mathbf{G}_T + \mathbf{M}_{d,T} \tag{5.90}$$

Putting everything together, we finally arrive at the expression:

$$\mathbf{MSE}_T(\mathbf{X}_{1:T}) = \mathbf{M}_{q,T} + \mathbf{b}_T^{\star}(\mathbf{X}_{1:T})\mathbf{b}_T^{\star}(\mathbf{X}_{1:T})^{\top} \tag{5.91}$$

where the optimal bias $\mathbf{b}_T^{\star}(\mathbf{X}_{1:T})$ is recursively computed from (5.82) and $\mathbf{M}_{q,k}$ from (5.90). While requiring significantly less computational burden than direct calculation by way of inverting $\mathbf{J}_{\mathbf{B},1:T}$, determination of this realization specific MSE matrix at each time consists of a number of steps. The complete approach for calculation at time $T$ based on the recursions developed is summarized as follows:

### 5.4.5 Application to a Maneuvering Target Tracking Problem

What follows is an application of the preceding results to a simple discrete-time 2-D target tracking problem. Let us first define the state vector, representing the target location and velocity, by the vector $\mathbf{X}_T = \begin{bmatrix} x_{1,T} & x_{2,T} & v_{1,T} & v_{2,T} \end{bmatrix}^\top$. Movement of the target at time $T$ is modeled by the transition equation:

$$\mathbf{X}_T = \mathbf{A}\mathbf{X}_{T-1} + \boldsymbol{\epsilon}_T \tag{5.92}$$

Where $\mathbf{A}$ is the state transition matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & T_0 & 0 \\ 0 & 1 & 0 & T_0 \\ 0 & 0 & T_0 & 0 \\ 0 & 0 & 0 & T_0 \end{bmatrix} \tag{5.93}$$

with $T_0$ is the step duration and $\boldsymbol{\epsilon}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_\epsilon)$ is the $(4 \times 1)$ state noise with:

$$\mathbf{Q}_\epsilon = \sigma_\epsilon^2 \begin{bmatrix} \frac{T_0^3}{3} & 0 & \frac{T_0^2}{2} & 0 \\ 0 & \frac{T_0^3}{3} & 0 & \frac{T_0^2}{2} \\ \frac{T_0^2}{2} & 0 & T_0 & 0 \\ 0 & \frac{T_0^2}{2} & 0 & T_0 \end{bmatrix} \tag{5.94}$$

Also, $\mathbf{X}_0 \sim \mathcal{N}\left(\begin{bmatrix} 0 & 0 & v_{1,0} & v_{2,0}, \end{bmatrix}^\top, \mathbf{Q}_\epsilon\right)$ corresponding to the target located initially about the origin, moving with constant velocity. A linear $4 \times 1$ observation vector is assumed as follows:

$$\mathbf{Y}_T = \mathbf{R}\mathbf{X}_T + \boldsymbol{\varsigma}_T \tag{5.95}$$

where $\boldsymbol{\varsigma}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_\varsigma)$ is the $4 \times 1$ observation noise, and $\mathbf{Q}_\varsigma$ is here assumed to be $\sigma_\varsigma^2\mathbf{I}$. For our example, we choose $\mathbf{R}$ to be a function of location only:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.96}$$

This is obviously a linear-Gaussian discrete-time filtering problem and it can easily be shown that (5.45) is satisfied; the Kalman filter is Bayes optimal and achieves the PCRB. The terms (5.59)-(5.61) are independent of time and have the simple form:

$$\begin{aligned} \mathbf{V}_T &= -\mathbf{A}^\top \mathbf{Q}_\epsilon^{-1} \\ \mathbf{U}_T &= \mathbf{A}^\top \mathbf{Q}_\epsilon^{-1} \mathbf{A} \\ \mathbf{W}_T &= \mathbf{Q}_\epsilon^{-1} \mathbf{A} + \mathbf{R}^\top \mathbf{Q}_\varsigma^{-1} \mathbf{R} \end{aligned}$$

The realization-dependent terms $\mathbf{Z}_{T-1}$ and $\mathbf{Z}_T$ are given by:

$$\mathbf{Z}_{T-1} = \mathbf{A}^\top \mathbf{Q}_\epsilon^{-1} \left( \mathbf{X}_T - \mathbf{A}\mathbf{X}_{T-1} \right)$$
$$\mathbf{Z}_T = -\mathbf{Q}_\epsilon^{-1} \left( \mathbf{X}_T - \mathbf{A}\mathbf{X}_{T-1} \right)$$

Recursive computation (not shown here) of the trajectory-specific MSE for the Bayes optimal estimator can then be carried out as outlined in the previous section.

To compactify presented simulation results, the sum of location-related terms for the PCRB and trajectory-specific MSE are shown rather than the individual components:

$$\widehat{\mathbf{MSE}}_T(\mathbf{X}_T) = \sum_{j=1}^{2} \left[ \mathbf{MSE}_T(\mathbf{X}_T) \right]_{j,j}$$
$$\widehat{\mathbf{PCRB}} = \sum_{j=1}^{2} \left[ \mathbf{PCRB} \right]_{j,j}$$

Figure 5.10 shows a single trajectory generated from the process equation with fixed $\sigma_\epsilon = 0.1$ and $\sigma_\varsigma = 0.1$. Figure 5.11 shows the theoretical $\widehat{\mathbf{MSE}}_T(\mathbf{X}_T)$ along with the experimental average error of the optimal Kalman filter for this trajectory. Note the dependence on $N$ of the experimental error; $N$ is the total number of trial runs for the Kalman filter where each trial consists of a different realization of the random observation trajectory. The absolute difference between the two is plotted in figure 5.12 for several values of $N$ where the convergence of experimental to theoretical error is clear. Also shown in figure 5.13 to demonstrate similar convergence is the bias for the first location component, $\left[ \mathbf{b}_T^\star(\mathbf{X}_{1:T}) \right]_1$.

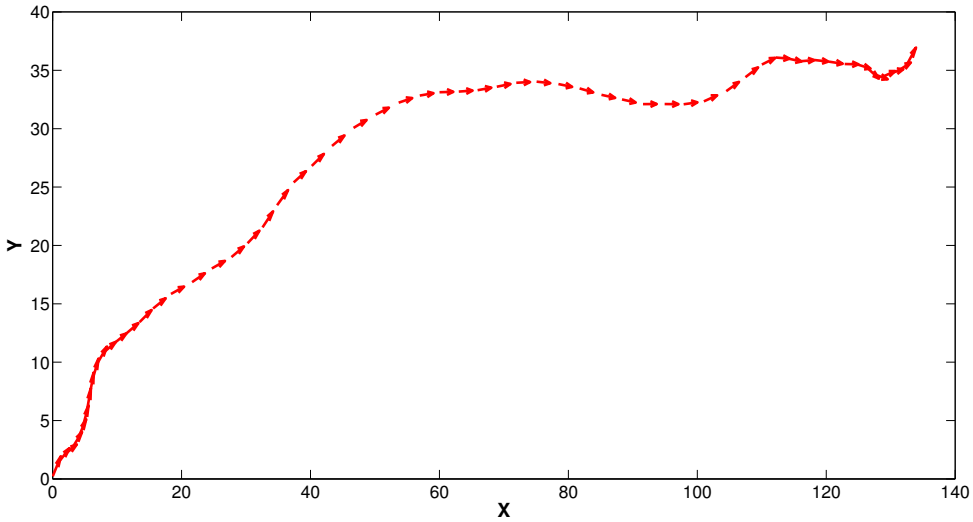


**Figure 5.10:** Single trajectory generated from the model with $\sigma_\epsilon = 0.1$ and $\sigma_\varsigma = 0.1$

Figure 5.15 shows $\widehat{\mathbf{PCRB}}$ for various values of $\sigma_\epsilon$ with $\sigma_\varsigma$ fixed at 0.1. Also plotted for each $\sigma_\epsilon$ is $\widehat{\mathbf{MSE}}_T(\mathbf{X}_T)$ for a number of trajectories randomly generated from the modeled state dynamics.

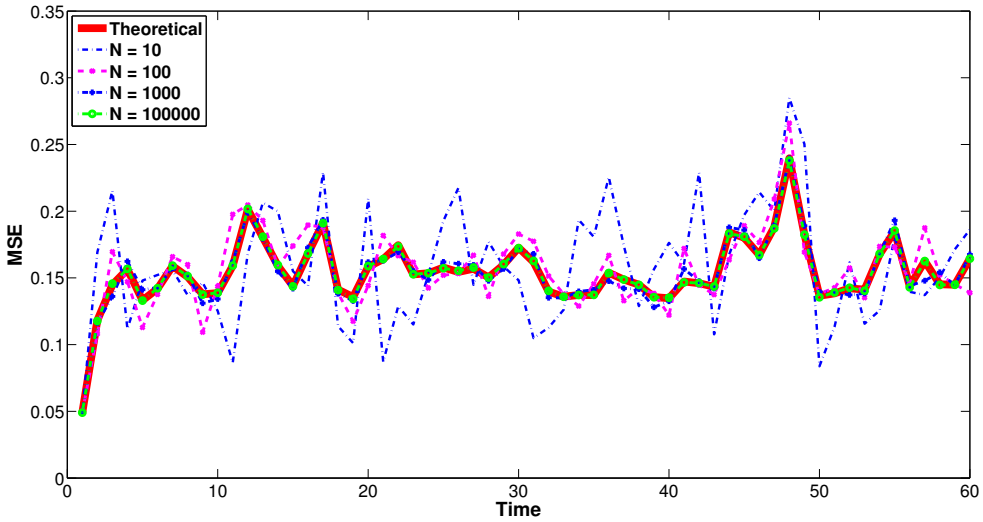


**Figure 5.11:** MSE of trajectory in figure 5.10 with $\sigma_\epsilon = 0.1$ and $\sigma_\varsigma = 0.1$

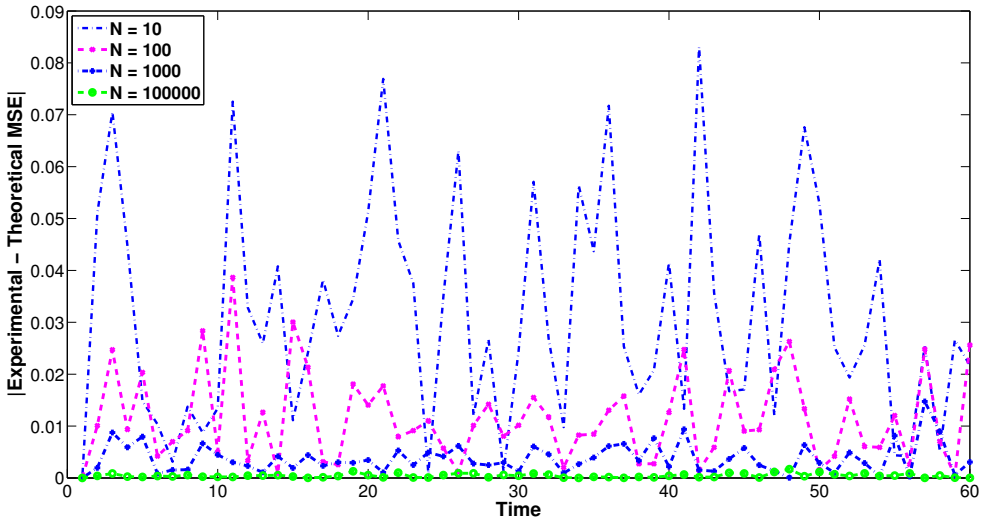


**Figure 5.12:** |Experimental - Theoretical| MSE for trajectory in figure 5.10 with $\sigma_\epsilon = 0.1$ and $\sigma_\varsigma = 0.1$

Since these trajectories obey the model, it is not unexpected that their specific MSE is close to the PCRB. Figure 5.14 shows the norm of the optimal bias,

$$\| \mathbf{b}_T^\star(\mathbf{X}_{1:T}) \| = \sum_{j=1}^{4} [\mathbf{b}_T^\star(\mathbf{X}_{1:T})]_j$$

for various assumed $\sigma_\epsilon$ but leaving this value fixed at $10^{-6}$ when generating the trajectories so as to clarify the comparison. Note how the bias norm significantly decreases as the assumed $\sigma_\epsilon$ increases; this is due to the estimator "shifting" weight at each $T$ from its prediction based on the prior estimate, to the incoming measurement.

The impact on $\widehat{\mathbf{MSE}}_T(\mathbf{X}_T)$ is next examined for trajectories which deviate from the assumed

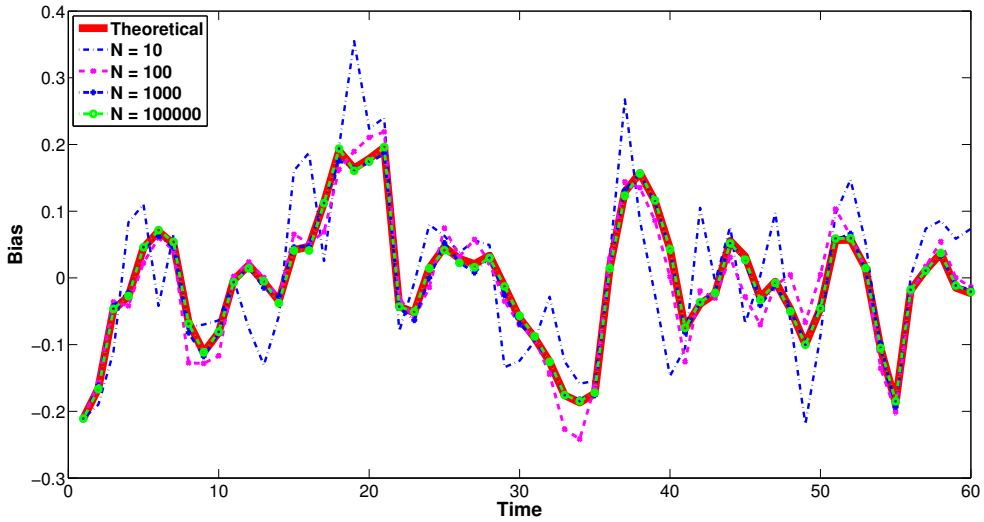


**Figure 5.13:** First location component of bias function for trajectory in figure 5.10 with $\sigma_\epsilon = 0.1$ and $\sigma_\varsigma = 0.1$

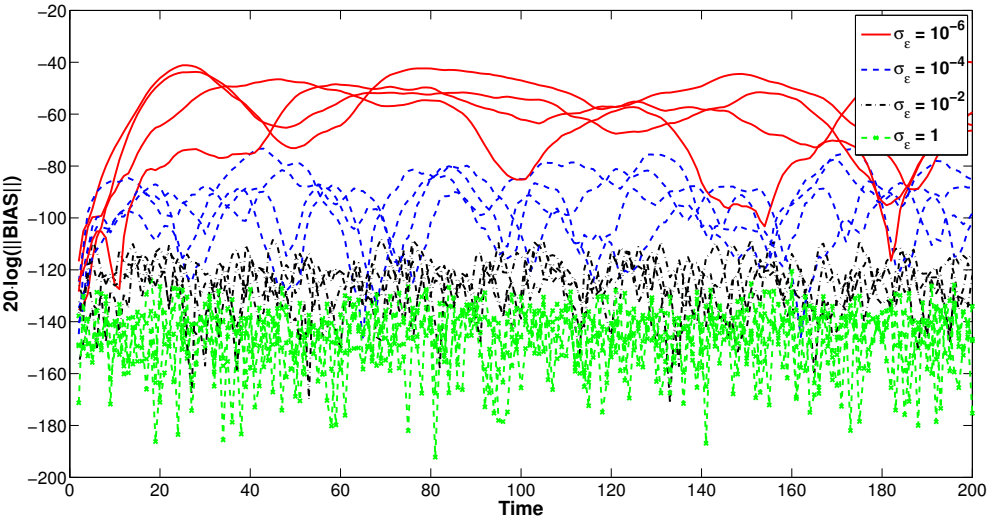


**Figure 5.14:** Logarithmic plot of $\| \mathbf{b}^\star_T(\mathbf{X}_{1:T}) \|$ for various assumed values of $\sigma_\epsilon$ with all trajectories generated using fixed $\sigma_\epsilon = 10^{-6}$

model dynamics. Specifically, let us consider trajectories $\check{\mathbf{X}}_T$ which initially follow the assumed process dynamics (straight-line motion) for some fixed time, then abruptly undergo a constant turn rate (CT) maneuver for a given duration, afterwards returning to the originally assumed dynamics for some time before the next maneuver. The model for a discretized CT maneuver as in [83] is adopted here. Note the estimated state vector is not modified here to include turn rate; this model serves only as a means to generate test trajectories and the actual estimation problem (assumed dynamics, measurement equation, noise statistics) remains the same. This test trajectory is then

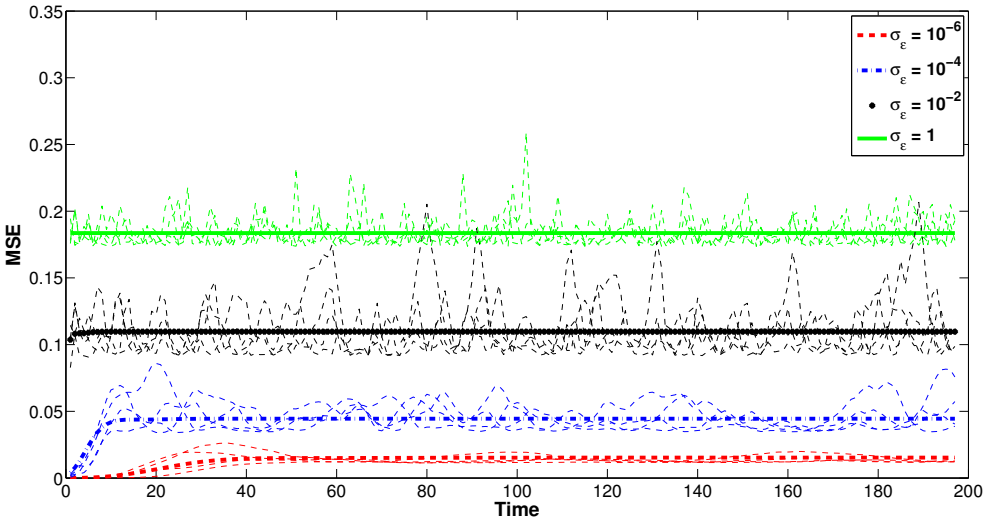


**Figure 5.15:** PCRB plotted for various $\sigma_\epsilon$. Heavy lines are the PCRB, thin dotted lines are the MSE for specific trajectories generated from the model.

described by

$$\check{\mathbf{X}}_T = \mathbf{F}\check{\mathbf{X}}_{T-1} + \epsilon_T$$

$$\mathbf{F} = \begin{cases} \mathbf{A} & \text{if } 0 \le T < T_{m,j} \text{ or } T_{m,j} + k_{m,j} < T < T_{m,j+1} \\ \mathbf{A_m} & \text{if } T_{m,j} \le T \le T_{m,j} + k_{m,j} \end{cases}$$

with

$$\mathbf{A_m} = \begin{bmatrix} 1 & 0 & \frac{\sin(\omega T_0)}{\omega} & -\frac{(1-\cos(\omega T_0))}{\omega} \\ 0 & 1 & \frac{(1-\cos(\omega T_0))}{\omega} & \frac{\sin(\omega T_0)}{\omega} \\ 0 & 0 & \cos(\omega T_0) & -\sin(\omega T_0) \\ 0 & 0 & \sin(\omega T_0) & \cos(\omega T_0) \end{bmatrix} \tag{5.97}$$

where $\omega$ is the turn rate, $T_{m,j}$ is the start time of the $j$-th maneuver, and $k_{m,j}$ is its duration.

Trajectories with two specific maneuvers are considered here, both with $\omega = 0.12$, one at $T = 50$ and another at $T = 140$, both lasting for 20 simulation steps. As done with trajectories matching the process dynamics, the PCRB and trajectory-specific MSE is evaluated for various values of $\sigma_\epsilon$. Simulations were first run (shown in figure 5.16 ) with $\sigma_\epsilon$ of a generated trajectory adjusted to match the assumed dynamics used for the estimator (i.e. $\widehat{\mathbf{MSE}}_T(\mathbf{X}_T)$ is calculated for trajectories generated with the same $\sigma_\epsilon$). It is interesting to note how the level of impact maneuvers have significantly changes with $\sigma_\epsilon$. While performance for small $\sigma_\epsilon$ is moderately better during the non-maneuvering phases, it is severely degraded during the maneuver. As $\sigma_\epsilon$ is increased, performance becomes insensitive to the maneuvers at the cost of a larger "steady-state" error.

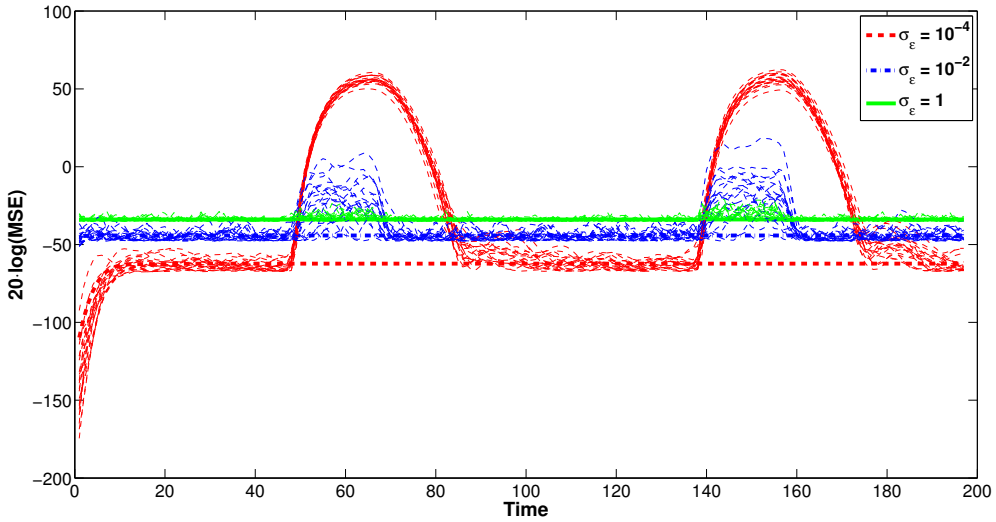


**Figure 5.16:** PCRB and MSE of the maneuvering trajectories plotted for various $\sigma_\epsilon$. Heavy lines are the PCRB, thin lines are trajectory-specific MSE. The assumed $\sigma_\epsilon$ matches actual $\sigma_\epsilon$ of the trajectories.

It can be shown in this specific example that further increases to $\sigma_\epsilon$ have a diminishing effect on this steady-state error and there is convergence to some level which is a function of the measurement noise; this can in a way be interpreted as "unbiasedness" for the filtering problem. Figure 5.17 shows sample trajectories generated for each $\sigma_\epsilon$ with height at each point as the value of $\widehat{\mathbf{MSE}}_T(\mathbf{X}_T)$.

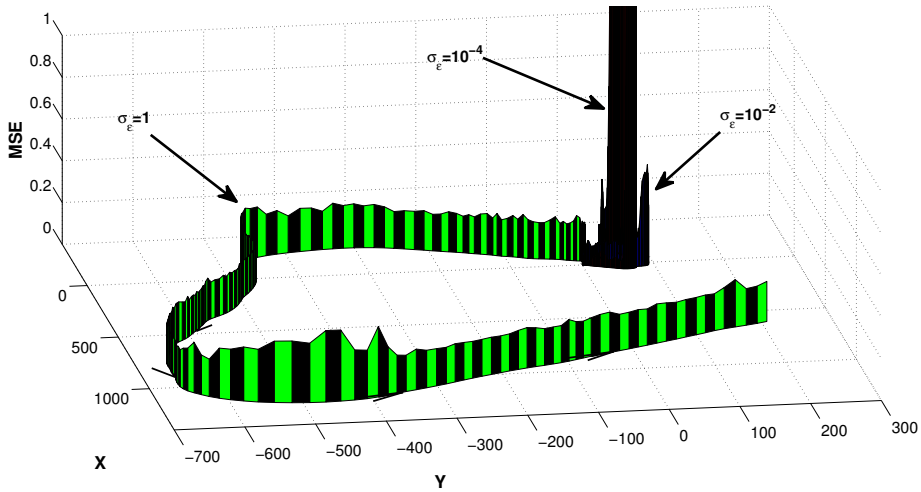


**Figure 5.17:** Sample trajectories with their respective MSE for various $\sigma_\epsilon$.

To get a clearer picture of figure 5.17 the same experiment is repeated, this time leaving $\sigma_\epsilon = 10^{-4}$ fixed during the generation of all the maneuvering trajectories and varying $\sigma_\epsilon$ only during calculation of $\mathbf{MSE}_T(\mathbf{X}_1 : T)$. Figures 5.18 and 5.19 show the corresponding results.

It is enlightening to examine the action of the bias during the trajectory maneuvers. Figure

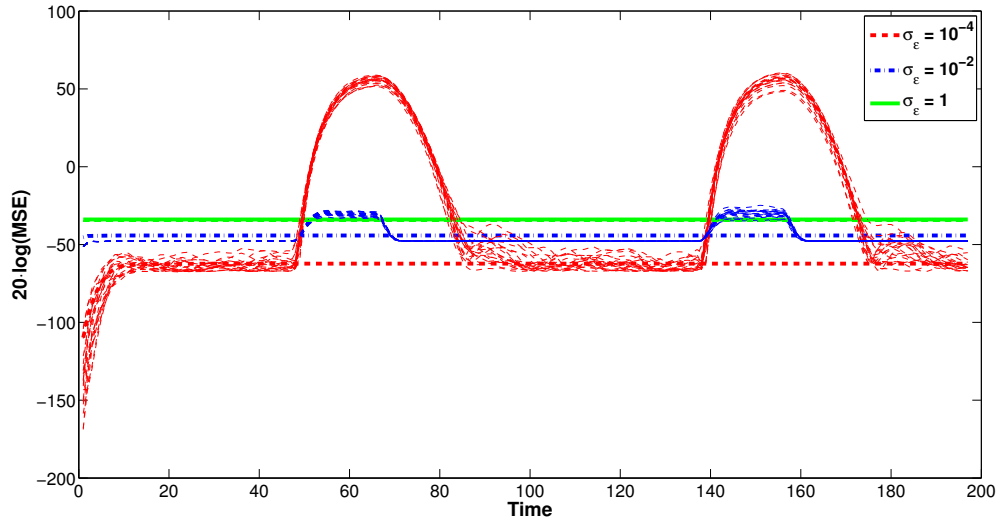


**Figure 5.18:** PCRB and MSE of the maneuvering trajectories plotted for various assumed $\sigma_\epsilon$. Heavy lines are the PCRB, thin lines are trajectory-specific MSE. All trajectories were generated with fixed $\sigma_\epsilon = 10^{-4}$

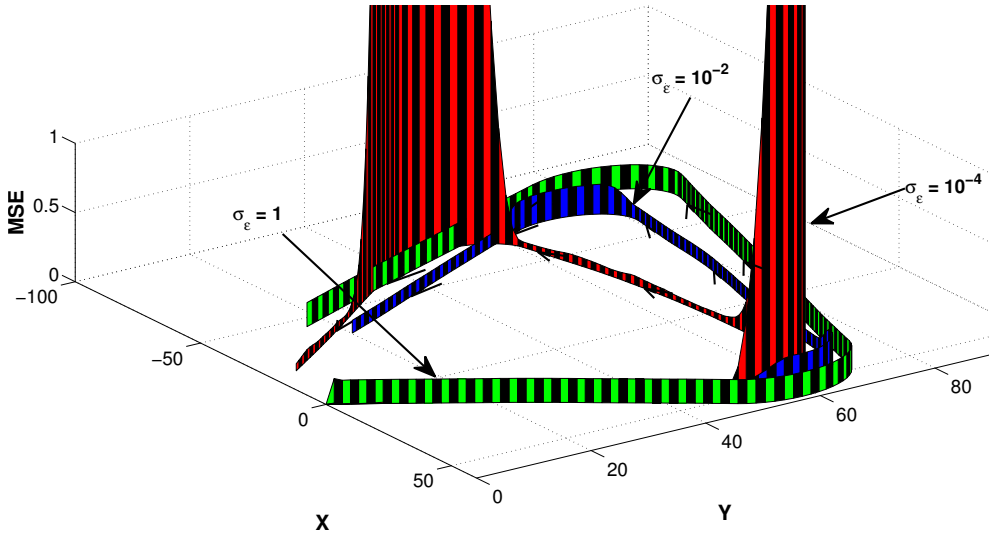


**Figure 5.19:** Sample trajectories with their respective MSE for various assumed $\sigma_\epsilon$ which is fixed at $10^{-4}$ during trajectory generation.

5.20 shows the bias norm for several trajectories generated with $\sigma_\epsilon$ fixed at $10^{-4}$. We immediately see that assuming a smaller $\sigma_\epsilon$ in the model dynamics imposes a dramatically larger bias norm during the maneuvers.

Figure 5.21 shows the bias vector alongside individual trajectories where we can clearly see the bias "acts to oppose" the change in trajectory; this "force" is "stronger" when $\sigma_\epsilon$ is assumed to be smaller.

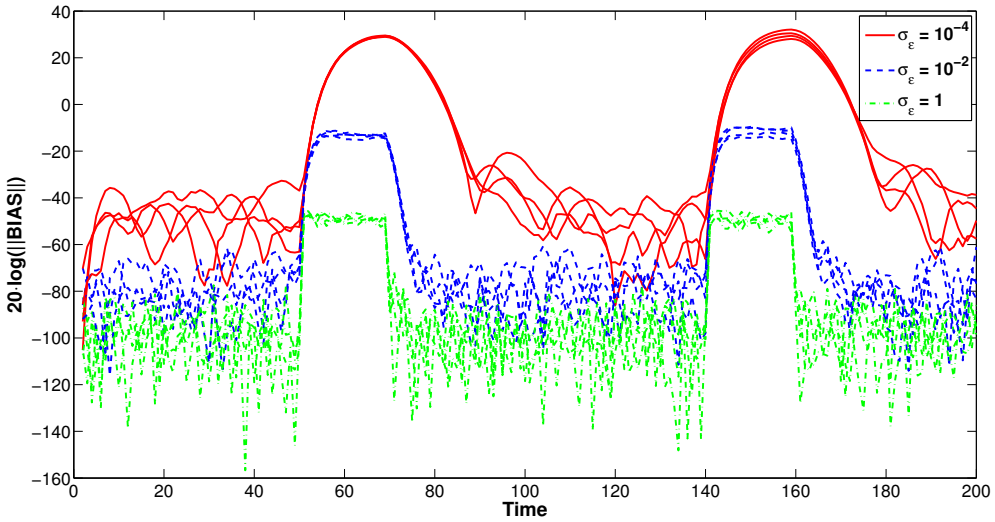


**Figure 5.20:** Logarithmic plot of $\| \mathbf{b}_T^\star(\mathbf{X}_{1:T}) \|$ for trajectories generated with $\sigma_\epsilon = 10^{-4}$. This value is again varied within the assumed model dynamics.

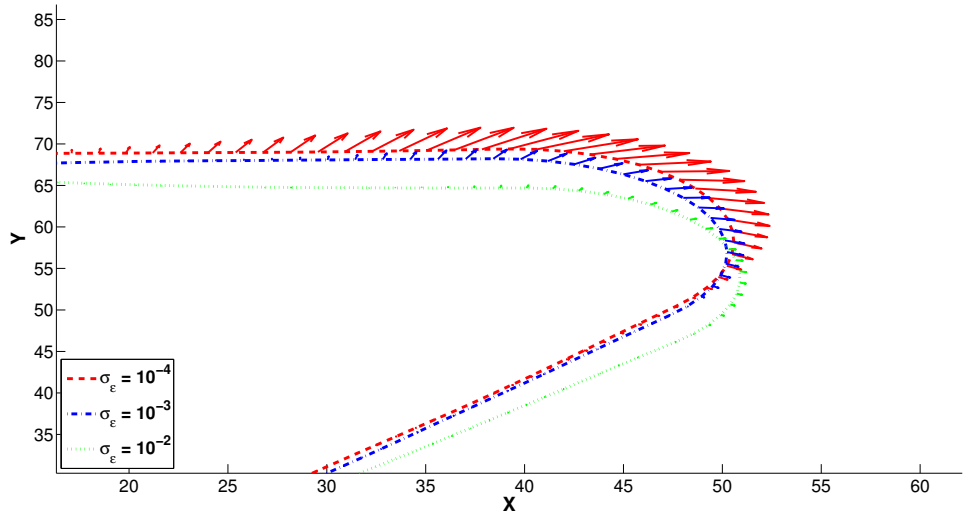


**Figure 5.21:** Sample trajectories with $\sigma_\epsilon$ fixed at $10^{-4}$ and varied only within model dynamics, plotted alongside location components of the bias vector (arrows).

## 5.5 Concluding Remarks

In this chapter a clear connection has been demonstrated relating optimal Bayesian estimation of a random parameter, to biased estimation of the same parameter when viewed as deterministic, for efficient estimation problems. The specific form of the bias function and resulting MSE for a given realization of the parameter was derived for scalar and vector parameters/observations. A recursive formula was then developed to allow for its application in the case of discrete-time filtering (i.e. target-tracking). Utilization of this method permitted a "deterministic" analysis to

be conducted within a Bayesian target tracking problem, offering insight into the performance of the Kalman filter conditioned on particular target trajectories. Such insight enabled easy evaluation of performance with trajectories falling outside the assumed model dynamics (maneuvering targets) and determination of the impact on performance exerted by changes to these model assumptions. An obvious avenue for further research on this topic involves the generalized extension to problems that do not satisfy the efficiency conditions (i.e. non-linear filter). While it is believed that progress in this area can be made by recasting the problem in the framework of Reproducing-Kernel-Hilbert-Space theory, this is left as an area open to future research.

# 6

# Investigation of RSS-Type Multi-Target Fusion Likelihood Function

This chapter details the investigation that has been conducted in pursuit of a tractable form for (2.37) using the specific measurement model outlined in (2.44); see the discussion surrounding these equations for the underlying motivation. To avoid obfuscation of the mathematical derivations presented in this section with the extended notation necessary for the methods presented earlier in the report, a relatively isolated notation is adopted here. Firstly, the dependence on $t$ in the subsequent chapter is dropped; all quantities are assumed to be for a given time. Recall that $\mathbf{Y}_{i,t}$ represents the set of measurements for all sensors assigned to agent $i$ at time $t$; here it is assumed that $K$ sensors produce measurements stacked into the vector $\mathbf{y}$ (the $i$-th element of this vector is $\mathbf{y}_i$ here). Additionally, the location of the $k$-th sensor is denoted as $\mathbf{q}_k$. The vector $\mathbf{X}_{\mathbf{S}_{i,t}}$ represents all targets of interest (TOIs) estimated by agent $i$; here without loss of generality it is assumed that there is a single TOI moving in the 2D plane, whose location coordinates are noted as $\mathbf{x}$. Finally, the vector $\mathbf{X}_{\mathbf{F}_{i,t}}$ represents the targets not estimated by agent $i$, but instead treated as interference; it is assumed there are a total of $L$ interferers with the location of the $i$-th interferer noted as $\mathbf{l}_{i,t}$.

It is desired to develop an expression for the joint conditional probability density of the vector random variable $\mathbf{y}$, given the location $\mathbf{x}$, whose individual components are generated as:

$$y_{\mathbf{l}} = h\left(\mathbf{x}\right) + \sum_{i=1}^{L} \frac{\Phi}{\|\mathbf{q}_k - \mathbf{l}_i\|^{\alpha} + \epsilon} + v_k \tag{6.1}$$

where $\epsilon$, and $\Phi$ are known constants, $v_k$ is independent Gaussian noise for the $k$-th sensor, and $\mathbf{l}_i$ is assumed to be distributed as a bivariate random variable, i.e., $\mathbf{l}_i \sim \mathcal{N}\left(\hat{\mathbf{l}}_i \sim \mathbf{C}_i\right)$, where $\mathbf{C}_i = \begin{bmatrix} \sigma_{1,i}^2 & \rho_t \sigma_{1,i}\sigma_{2,i} \\ \rho_t \sigma_{1,i}\sigma_{2,i} & \sigma_{2,i}^2 \end{bmatrix}$. Namely, the conditional distribution we wish to develop an expression

for is written as $f\left(\mathbf{y} \mid \mathbf{x}, \hat{\mathbf{l}}_{1:L}, \mathbf{C}_{1:L}\right)$.

Finding this density in the most general scenario, i.e., for multiple sensors and interferers, represents a formidable task and likely does not yield to a closed form. However, specific configurations have been investigated here successfully which will be incrementally presented.

### 6.0.1 Single Sensor Theoretical Results

Let us assume there exists a single sensor measurement, $y$ located at $\mathbf{q}$, along with a single interfering source, $\mathbf{l}$. Also, we let us remove the contribution of sensor noise from (6.1) (set $v_k = 0$ here and focus on the contribution due to the interfering target, denoted as $y_{\mathbf{l}}$ and given by,

$$y_k = \frac{\Phi}{\|\mathbf{q} - \mathbf{l}\|^{\alpha} + \epsilon} \tag{6.2}$$

By introducing the auxiliary variable, $\theta$, the coordinates of $\mathbf{l} = [l_1 \; l_2]^{\top}$ can be written as

$$\begin{aligned} \mathbf{l} &= \mathbf{g}(y_{\mathbf{l}}, \theta) \\ &= \left(\frac{\Phi}{y_{\mathbf{l}}} - \epsilon\right)^{\frac{1}{\alpha}} \begin{bmatrix}\cos\theta \\ \sin\theta\end{bmatrix} + \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}. \end{aligned} \tag{6.3}$$

Since there is a clearly invertible mapping between the pair $(y_{\mathbf{l}}, \theta)$ and $\mathbf{l}$, the method of transformation of random variables can be applied to yield the expression

$$\begin{aligned} f\left(y_{\mathbf{l}}, \theta \mid \hat{\mathbf{l}}, \mathbf{C}\right) &= |\det\ (\mathbf{J}(\mathbf{l}))|_{\mathbf{l}=\mathbf{g}(y_{\mathbf{l}},\theta)} \\ &\quad \times f\left(\mathbf{g}(y_{\mathbf{l}}, \theta) \mid \hat{\mathbf{l}}, \hat{\mathbf{C}}\right), \end{aligned} \tag{6.4}$$

where $|\mathbf{J}(\mathbf{l})|$ is the Jacobian of the transformation from $(y_{\mathbf{l}}, \theta)$ to $\mathbf{l}$. For convenience, let us make the following definition:

$$\eta_{y_{\mathbf{l}}} = \frac{\Phi}{y_{\mathbf{l}}} - \epsilon. \tag{6.5}$$

Allowing us to write,

$$\begin{aligned} \det\ (\mathbf{J}(\mathbf{l})) &= \begin{vmatrix} \frac{\partial l_1}{\partial y_{\mathbf{l}}} & \frac{\partial l_1}{\partial \theta} \\ \frac{\partial l_2}{\partial y_{\mathbf{l}}} & \frac{\partial l_2}{\partial \theta} \end{vmatrix} \\ &= \begin{vmatrix} \eta_{y_{\mathbf{l}}}^{\left(\frac{1}{\alpha}-1\right)} \left(\frac{-\Phi}{\alpha y_{\mathbf{l}}^2}\right) \cos(\theta) & -\eta_{y_{\mathbf{l}}}^{\frac{1}{\alpha}} \sin(\theta) \\ \eta_{y_{\mathbf{l}}}^{\left(\frac{1}{\alpha}-1\right)} \left(\frac{-\Phi}{\alpha y_{\mathbf{l}}^2}\right) \sin(\theta) & \eta_{y_{\mathbf{l}}}^{\frac{1}{\alpha}} \cos(\theta) \end{vmatrix} \\ &= \frac{\Phi}{\alpha y_{\mathbf{l}}^2} \eta_{y_{\mathbf{l}}}^{\left(\frac{2}{\alpha}-1\right)}. \end{aligned} \tag{6.6}$$

After further introducing the auxiliary random vector $\mathbf{d}_t = \begin{bmatrix}\mathbf{l}_1 - \mathbf{q}_1 \\ \mathbf{l}_2 - \mathbf{q}_2\end{bmatrix}$, the measurement component $y_{\mathbf{l}_t}$ can be rewritten as

$$y_{\mathbf{l}} = \frac{\Phi}{(\mathbf{d}^{\top}\mathbf{d})^{\frac{\alpha}{2}} + \epsilon}, \tag{6.7}$$

where $\mathbf{d} \sim \mathcal{N}\left(\hat{\mathbf{d}}, \hat{\mathbf{C}}\right)$ with $\hat{\mathbf{d}} = \begin{bmatrix} \hat{\mathbf{l}}_1 - \mathbf{q}_1 \\ \hat{\mathbf{l}}_2 - \mathbf{q}_2 \end{bmatrix}$. By making use of equations (6.3)-(6.7), we obtain the joint density of the pair $(y_{\mathbf{l}}, \theta)$ in the form

$$f\left(y_{\mathbf{l}}, \theta \mid \hat{\mathbf{l}}, \mathbf{C}\right) = \left[\frac{\Phi}{\alpha y_{\mathbf{l}}^2} \eta_{y_{\mathbf{l}}}^{\left(\frac{2}{\alpha}-1\right)}\right] \mathcal{N}\left(\eta_{y_{\mathbf{l}}}^{\frac{1}{\alpha}} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \,\middle|\, \hat{\mathbf{d}}, \hat{\mathbf{C}}\right), \tag{6.8}$$

where $\mathcal{N}\left(\star \mid \hat{\mathbf{d}}, \mathbf{C}\right)$ denotes that the $2 \times 1$ vector argument symbolized with a $\star$ has a bivariate normal probability density with mean $\hat{\mathbf{d}}$ and covariance matrix $\mathbf{C}$.

### 6.0.2 Solution Method 1

The desired expression is obtained by marginalizing $\theta$, i.e., the bivariate normal density needs to be integrated along the circle centered at the origin with radius $\eta_{y_{\mathbf{l}}}^{\frac{1}{\alpha}}$. This integral has been well-studied under various contexts, with related literature found in [84], [85], and [86]. The most relevant study appears in [87], where a closed form solution involving an infinite series of Bessel function products is presented for the case where the individual components of the random vector have different variances, but are uncorrelated. A more general solution, allowing for nonzero correlation between components, can be obtained by noting that,

$$\mathcal{N}\left(\eta_{y_{\mathbf{l}}}^{\frac{1}{\alpha}} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \,\middle|\, \mathbf{d}, \mathbf{C}\right) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left[\mathcal{T}_1\left(\eta_{y_{\mathbf{l}}}, \theta, \hat{\mathbf{d}}, \mathbf{C}\right)\right], \tag{6.9}$$

with,

$$\mathcal{T}_1\left(\eta_{y_{\mathbf{l}}}, \theta, \hat{\mathbf{d}}, \mathbf{C}\right) = -\frac{1}{2\left(1-\rho^2\right)} \left( k_1 + \left(k_2 - \sqrt{k_5^2 + k_6^2}\right) \eta_{y_{\mathbf{l}}}^{\frac{2}{\alpha}} + \left(\sqrt{k_3^2 + k_4^2}\right) \eta_{y_{\mathbf{l}}}^{\frac{1}{\alpha}} \cos\left(\theta - \phi_1\right) \right.$$
$$\left. + 2\left(\sqrt{k_5^2 + k_6^2}\right) \eta_{y_{\mathbf{l}}}^{\frac{2}{\alpha}} \cos^2\left(\theta - \phi_2\right) \right), \tag{6.10}$$

where the constants and phases are defined as

$$\begin{aligned} k_1 &= \frac{\hat{d}_1^2}{\sigma_1^2} + \frac{\hat{d}_2^2}{\sigma_2^2} - \frac{2\rho_t \hat{d}_1 \hat{d}_2}{\sigma_1\sigma_2}, & k_2 &= \frac{1}{2}\left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right), \\ k_3 &= 2\left(\frac{\rho_t \hat{d}_2}{\sigma_1\sigma_2} - \frac{\hat{d}_1}{\sigma_1^2}\right), & k_4 &= 2\left(\frac{\rho \hat{d}_1}{\sigma_1\sigma_2} - \frac{\hat{d}_2}{\sigma_2^2}\right), \\ k_5 &= \frac{1}{2}\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right), & k_6 &= -\frac{\rho}{\sigma_1\sigma_2}, \\ \phi_1 &= \text{atan2}\left(\frac{k_4}{k_3}\right), & \phi_2 &= \frac{1}{2}\text{atan2}\left(\frac{k_6}{k_5}\right). \end{aligned} \tag{6.11}$$

With the argument in this form, the result derived in [88] can be leveraged,

$$\int_0^{2\pi} e^{in\theta} \exp\left[a\cos\left(\theta - \alpha\right) + 2b\cos^2\left(\theta - \beta\right)\right] d\theta = 2\pi e^b e^{in\alpha} \sum_{j=-\infty}^{\infty} e^{2ij(\alpha-\beta)} \mathrm{I}_{2j+n}\left(a\right) \mathrm{I}_j\left(b\right), \tag{6.12}$$

where $i = \sqrt{-1}$ and $\mathrm{I}_j(\cdot)$ is the $j$th order Modified Bessel function of the first kind. By matching the corresponding terms in (6.12) with those in the integral, and noting that for $j$ an integer, $\mathrm{I}_{-j}(x) = \mathrm{I}_j(x)$, and $\mathrm{I}_{2j}(-x) = \mathrm{I}_{2j}(x)$, the final form is obtained,

$$f_{y_1}\left(y_1 \mid \hat{\mathbf{l}}, \mathbf{C}\right) = \frac{\Phi \eta_{y_1}^{\left(\frac{2}{\alpha}-1\right)}}{\alpha\sigma_1\sigma_2\left(\sqrt{1-\rho^2}\right)y_1^2} e^{-\frac{k_1+k_2\eta_{y1}^{\frac{2}{\alpha}}}{2(1-\rho^2)}} \times \left\{ \mathrm{I}_0\left(\frac{-\sqrt{k_5^2+k_6^2}}{2\left(1-\rho_t^2\right)}\eta_{y_1}^{\frac{2}{\alpha}}\right)\mathrm{I}_0\left(\frac{\sqrt{k_3^2+k_4^2}}{2\left(1-\rho^2\right)}\eta_{y_1}^{\frac{1}{\alpha}}\right) + 2\sum_{j=1}^{\infty}\left[ \mathrm{I}_j\left(\frac{-\sqrt{k_5^2+k_6^2}}{2\left(1-\rho_t^2\right)}\eta_{y_1}^{\frac{2}{\alpha}}\right) \times \mathrm{I}_{2j}\left(\frac{\sqrt{k_3^2+k_4^2}}{2\left(1-\rho^2\right)}\eta_{y_1}^{\frac{1}{\alpha}}\right)\cos\left(2j\left(\phi_1-\phi_2\right)\right)\right]\right\}, \quad (6.13)$$

with $\eta_{y_1}$ as defined in (6.5) and the parameters $k_1$-$k_6$ and phases $\phi_1$, $\phi_2$, defined in (6.11).

### 6.0.3 Solution Method 2

An alternative expression for (6.13) in the case where $\rho = 0$ [1] can be found by rewriting equation (6.8) as,

$$\begin{aligned} f(y_1,\theta) &= \frac{\Phi\eta_{y_k}^{\frac{2}{\alpha}-1}}{2\pi\alpha\sigma_1\sigma_2 y_k^2}\exp\left[-\frac{1}{2\sigma_1^2}\left(\eta_{y_k}^{\frac{1}{\alpha}}\cos\theta-\hat{d}_1\right)^2-\frac{1}{2\sigma_2^2}\left(\eta_{y_k}^{\frac{1}{\alpha}}\sin\theta-\hat{d}_2\right)^2\right] \\ &= \frac{\Phi\eta_{y_k}^{\frac{2}{\alpha}-1}}{2\pi\alpha\sigma_1\sigma_2 y_k^2} e^{\left[-\frac{1}{2}\left(\frac{\hat{d}_1^2}{\sigma_1^2}+\frac{\hat{d}_2^2}{\sigma_2^2}\right)\right]} e^{\left[-\frac{1}{2\sigma_1^2}\left(\eta_{y_k}^{\frac{2}{\alpha}}\cos^2\theta-2\eta_{y_k}^{\frac{1}{\alpha}}\hat{d}_1\cos\theta\right)-\frac{1}{2\sigma_2^2}\left(\eta_{y_k}^{\frac{2}{\alpha}}\sin^2\theta-2\eta_{y_k}^{\frac{1}{\alpha}}\hat{d}_2\sin\theta\right)\right]} \end{aligned} \quad (6.14)$$

Simplifying the argument of the second exponential term produces:

$$\begin{aligned} &-\frac{1}{2\sigma_1^2}\left(\eta_{y_k}^{\frac{2}{\alpha}}\cos^2\theta-2\eta_{y_k}^{\frac{1}{\alpha}}\hat{d}_1\cos\theta\right)-\frac{1}{2\sigma_2^2}\left(\eta_{y_k}^{\frac{2}{\alpha}}\sin^2\theta-2\eta_{y_k}^{\frac{1}{\alpha}}\hat{d}_2\sin\theta\right) \\ &= -\frac{\sigma_2^2-\sigma_1^2}{2\sigma_1^2\sigma_2^2}\eta_{y_k}^{\frac{2}{\alpha}}\cos^2\theta+\eta_{y_k}^{\frac{1}{\alpha}}\left(\frac{\hat{d}_1}{\sigma_1^2}\cos\theta+\frac{\hat{d}_2}{\sigma_2^2}\sin\theta\right)-\frac{\eta_{y_k}^{\frac{2}{\alpha}}}{2\sigma_2^2} \end{aligned} \quad (6.15)$$

This can be expressed in the form,

$$f(y_1,\theta) = \frac{\Phi\eta_{y_k}^{\frac{2}{\alpha}-1}}{2\pi\alpha\sigma_1\sigma_2 y_k^2} e^{\left[-\frac{1}{2}\left(\frac{\hat{d}_1^2}{\sigma_1^2}+\frac{\hat{d}_2^2+k^2}{\sigma_2^2}\right)\right]} e^{\left[\varrho_1\cos^2\theta+\varrho_2\cos\theta+\varrho_3\sin\theta\right]} \quad (6.16)$$

with,

$$\varrho_1 = -\frac{\sigma_2^2-\sigma_1^2}{2\sigma_1^2\sigma_2^2}\eta_{y_k}^{\frac{2}{\alpha}} \quad , \quad \varrho_2 = \eta_{y_k}^{\frac{1}{\alpha}}\frac{\hat{d}_1}{\sigma_1^2} \quad , \quad \varrho_3 = \eta_{y_k}^{\frac{1}{\alpha}}\frac{\hat{d}_2}{\sigma_2^2} \quad (6.17)$$

[1] This can likely be extended to arbitrary $\rho$ but was not attempted here

To marginalize out $\theta$, one can proceed with computation of the integral $\int_0^{2\pi} e^{\varrho_1 \cos^2\theta + \varrho_2 \cos\theta + \varrho_3 \sin\theta} d\theta$ by first expanding the exponential into an infinite series, using the identity $e^x = \sum_0^\infty \frac{x^n}{n!}$. This produces:

$$e^{\varrho_1 \cos^2\theta + \varrho_2 \cos\theta + \varrho_3 \sin\theta} = \sum_{n=0}^{\infty} \frac{\left(\varrho_1 \cos^2\theta + \varrho_2 \cos\theta + \varrho_3 \sin\theta\right)^n}{n!} \tag{6.18}$$

which can be reduced further by iteratively applying the binomial theorem. Letting $a = \varrho_3 \sin\theta$, $b = \varrho_1 \cos^2\theta$, $c = \varrho_2 \cos\theta$, and $d = b + c$:

$$\begin{aligned}
\sum_{n=0}^{\infty} \frac{(a+d)^n}{n!} &= \sum_{n=0}^{\infty} \frac{1}{n!} \left[ \sum_{k=0}^{n} \binom{n}{k} a^{n-k} d^k \right] = \\
&= \sum_{n=0}^{\infty} \frac{1}{n!} \left[ \sum_{k=0}^{n} \binom{n}{k} a^{n-k} (b+c)^k \right] \\
&= \sum_{n=0}^{\infty} \frac{1}{n!} \left[ \sum_{k=0}^{n} \binom{n}{k} a^{n-k} \left[ \sum_{l=0}^{k} \binom{k}{l} b^{k-l} c^l \right] \right] \\
&= \sum_{n=0}^{\infty} \sum_{k=0}^{n} \sum_{l=0}^{k} \frac{a^{n-k} b^{k-l} c^l}{l!(n-k)!(k-l)!} \\
&= \sum_{n=0}^{\infty} \sum_{k=0}^{n} \sum_{l=0}^{k} \frac{\varrho_3^{n-k} \varrho_2^l \varrho_1^{k-l}}{l!(n-k)!(k-l)!} \sin^{n-k}\theta \cos^{2k-l}\theta
\end{aligned} \tag{6.19}$$

It is then necessary to integrate this expression over $\theta$ for each term in the summation. Isolating the terms that depend on $\theta$ gives way to the integral, $\int_0^{2\pi} \sin^{n-k}\theta \cos^{2k-l}\theta d\theta$. Setting $j = n - k$ and $h = 2k - l$, this has the form,

$$\int_0^{2\pi} \sin^j\theta \cos^h\theta d\theta \tag{6.20}$$

With odd $j$ and even $h$, $\sin^j\theta$ is anti-symmetric about $\theta = \pi$ while $\cos^h\theta$ is symmetric about $\theta = \pi$, thus the integrand is anti-symmetric about $\theta = \pi$ for odd $j$ and the integral over any interval centered at $\theta = \pi$ vanishes . If $h$ is odd, the integrand is anti-symmetric about $\theta = \frac{\pi}{2}$ over the interval $[0, \pi]$ and anti-symmetric about $\theta = \frac{3\pi}{2}$ over the interval $[\pi, 2\pi]$; by splitting the original integration over these two intervals it is obvious the result again vanishes. We are thus left with the only non-zero case; when both $j$ and $h$ are even. Making repeated use of the following two power-reduction formulae,

$$\int_a^b \sin^j\theta \cos^h\theta d\theta = -\left.\frac{\sin^{j-1}\theta \cos^{h+1}\theta}{(j+h)}\right|_a^b + \frac{j-1}{j+h} \int_a^b \sin^{j-2}\theta \cos^h\theta d\theta \tag{6.21}$$

$$\int_a^b \cos^h\theta d\theta = \left.\frac{\sin\theta \cos^{h-1}\theta}{h}\right|_a^b + \frac{h-1}{h} \int_a^b cos^{h-2}\theta d\theta \tag{6.22}$$

It can be seen that (6.20) can be reduced as follows:

$$\int_0^{2\pi} \sin^j\theta\cos^h\theta d\theta = \left(\frac{j-1}{j+h}\right)\left(\frac{j-3}{j+h-2}\right)\left(\frac{j-5}{j+h-4}\right)\ldots\left(\frac{1}{h+2}\right)\int_0^{2\pi}\cos^h\theta d\theta$$
$$= \frac{[(j-1)(j-3)\ldots 1][(h-1)(h-3)\ldots 1]}{(j+h)(j+h-2)(j+h-4)\ldots(h)(h-2)\ldots(2)}(2\pi) \tag{6.23}$$

Noting that for even $j$ and $h$ one has,

$$\Gamma(\frac{j+1}{2}) = \left(\frac{j-1}{2}\right)\left(\frac{j-3}{2}\right)\ldots\left(\frac{1}{2}\right)\Gamma(\frac{1}{2})$$
$$= \sqrt{\pi}\left(\frac{1}{2}\right)^{\frac{j}{2}}(j-1)(j-3)\ldots(1) \tag{6.24}$$

and,

$$\left(\frac{j+h}{2}\right)! = \left(\frac{j+h}{2}\right)\left(\frac{j+h}{2}-1\right)\left(\frac{j+h}{2}-2\right)\ldots\left(\frac{h}{2}\right)\left(\frac{h-2}{2}-1\right)\ldots(1)$$
$$= \left(\frac{1}{2}\right)^{\frac{j+h}{2}}(j+h)(j+h-2)\ldots(h)(h-2)\ldots(2) \tag{6.25}$$

Using (6.24) for each term in the numerator and (6.25) for the term in the denominator of (6.23), one arrives at:

$$\int_0^{2\pi}\sin^j\theta\cos^h\theta d\theta = 2\frac{\Gamma\left(\frac{j+1}{2}\right)\Gamma\left(\frac{h+1}{2}\right)}{\Gamma\left(\frac{h+j}{2}+1\right)} = 2\mathbf{B}\left(\frac{j+1}{2},\frac{h+1}{2}\right) \tag{6.26}$$

where $\mathbf{B}(x,y)$ denotes the complete Beta function. To use (6.26) in (6.19) it is first recognized that for a given term in the summation to be non-zero, $l$ must be even and $n$ and $k$ must both be either even or odd. This constraint can be achieved with the qualifying expression,

$$\delta_{(n,k,l)} \triangleq \frac{1}{4}\left[1+(-1)^l\right]\left[(-1)^{n+k}+(-1)^{2k}\right] = \begin{cases}1 & \text{if } l \textit{ even and } n,k \textit{ both even or odd}\\ 0 & \text{else}\end{cases} \tag{6.27}$$

As a result, it can now be written:

$$L[\varrho_1,\varrho_2,\varrho_3] \triangleq \int_0^{2\pi} e^{\varrho_1\cos^2\theta+\varrho_2\cos\theta+\varrho_3\sin\theta}d\theta =$$
$$\sum_{n=0}^{\infty}\sum_{k=0}^{n}\sum_{l=0}^{k}\frac{2\delta_{(n,k,l)}\varrho_3^{n-k}\varrho_2^l\varrho_1^{k-l}}{l!(n-k)!(k-l)!}\mathbf{B}\left(\frac{n-k+1}{2},\frac{2k-l+1}{2}\right) \tag{6.28}$$

Substituting the arguments for $L[\varrho_1,\varrho_2,\varrho_3]$ as defined in (6.17) allows the sum to be expressed as:

$$L\left[-\frac{\sigma_2^2-\sigma_1^2}{2\sigma_1^2\sigma_2^2}\eta_{y_k}^{\frac{2}{\alpha}},\frac{\hat{d}_1}{\sigma_1^2}\eta_{y_k}^{\frac{1}{\alpha}},\frac{\hat{d}_2}{\sigma_2^2}\eta_{y_k}^{\frac{1}{\alpha}}\right] =$$
$$\sum_{n=0}^{\infty}\sum_{k=0}^{n}\sum_{l=0}^{k}\left(\frac{2\eta_{y_1}^{\frac{n+k-l}{\alpha}}\delta_{(n,k,l)}}{l!(n-k)!(k-l)!}\left(-\frac{\sigma_2^2-\sigma_1^2}{2\sigma_1^2\sigma_2^2}\right)^{k-l}\left(\frac{\hat{d}_1}{\sigma_1^2}\right)^l\left(\frac{\hat{d}_2}{\sigma_2^2}\right)^{n-k}\mathbf{B}\left(\frac{n-k+1}{2},\frac{2k-l+1}{2}\right)\right) \tag{6.29}$$

Recognizing that $\frac{n+k-l}{2}$ is always an integer, it is noted that truncating the outer summation over $n$ to $N$ terms results in a polynomial of order $N$ with respect to $\eta_{y_k}^{\frac{2}{\alpha}}$. This observation was used to develop a feasible algorithm for approximating the series to high numerical precision and is discussed further in section 6.1.

Using equations (6.16),(6.17), and (6.28) produces for $f\left(y_{\mathbf{I}} \mid \hat{\mathbf{I}}, \mathbf{C}\right)$:

$$f(y_{\mathbf{I}} \mid \hat{\mathbf{I}}, \mathbf{C}) = \left( \frac{\Phi \eta_{y_k}^{\frac{2}{\alpha}-1}}{2\pi\alpha\sigma_1\sigma_2 y_k^2} \exp\left( -\frac{1}{2}\left( \frac{\hat{d}_1^2}{\sigma_1^2} + \frac{\hat{d}_2^2 + \eta_{y_k}^{\frac{2}{\alpha}}}{\sigma_2^2} \right) \right) L\left[ -\frac{\sigma_2^2 - \sigma_1^2}{2\sigma_1^2\sigma_2^2}\eta_{y_k}^{\frac{2}{\alpha}}, \frac{\hat{d}_1}{\sigma_1^2}\eta_{y_k}^{\frac{1}{\alpha}}, \frac{\hat{d}_2}{\sigma_2^2}\eta_{y_k}^{\frac{1}{\alpha}} \right] \right) \tag{6.30}$$

For the most part, this form has been found to yield inferior numerical convergence results to that of the first solution due to the doubly-nested summation, which can require an excessively large number of terms to achieve a given precision. However, as will be shown, there are specific parameter values that do not allow the first solution to be accurately computed easily due to the numerical precision required. In that case, this form does provide a viable alternative.

## 6.1 Single Sensor Numerical Results

Due to the presence of Bessel functions in the expression, the ability to evaluate (6.13) accurately is heavily dependent on the specific values of parameters in the model, namely the values for $k_1$ - $k_6$ defined in (6.11). While fast convergence with excellent accuracy is achieved for smaller parameter values, larger values for these parameters generate more numerical difficulty owing to the resulting large-argument Bessel function computation that is necessary. In certain cases, equation(6.13) becomes nearly impossible to compute to reasonable precision and one must resort to the alternative form given if this accuracy is required.

What follows is a comparison between (6.13) when $\rho = 0$ and the empirical density $\widehat{f}\left(y_{\mathbf{I}} \mid \hat{\mathbf{I}}, \mathbf{C}\right)$ which is generated by first drawing samples of $\hat{\mathbf{I}}$, then computing the corresponding sample of $y_{\mathbf{I}}$ using (6.2), and generating a histogram plot of the resulting sample set. Specific parameter values have been selected to demonstrate the numerical accuracy that can be achieved, along with the varying shape that this expression takes on depending on the parameters. In all cases, the number of samples drawn is set to $N = 5 \times 10^5$, the number of bins for the empirical histogram is set to 200, the sensor is located at the origin, $\Phi = 10$, and $\alpha = 2$,

### 6.1.1 Large Estimate Variances

The situation with $\hat{\mathbf{I}}$ close to the sensor is first examined; in this case there is a high likelihood that many of the target samples drawn will yield very high values for the sensor measurement, limited

only by $\epsilon$. As such, one can expect the shape of the density to be highly sensitive to this parameter. Figure 6.1 shows one example scenario with $\bar{\mathbf{l}} = \begin{bmatrix} -0.2 & 0.1 \end{bmatrix}^\top$ and $\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}$, where the sensor location and drawn target (interferer) estimate samples are plotted.

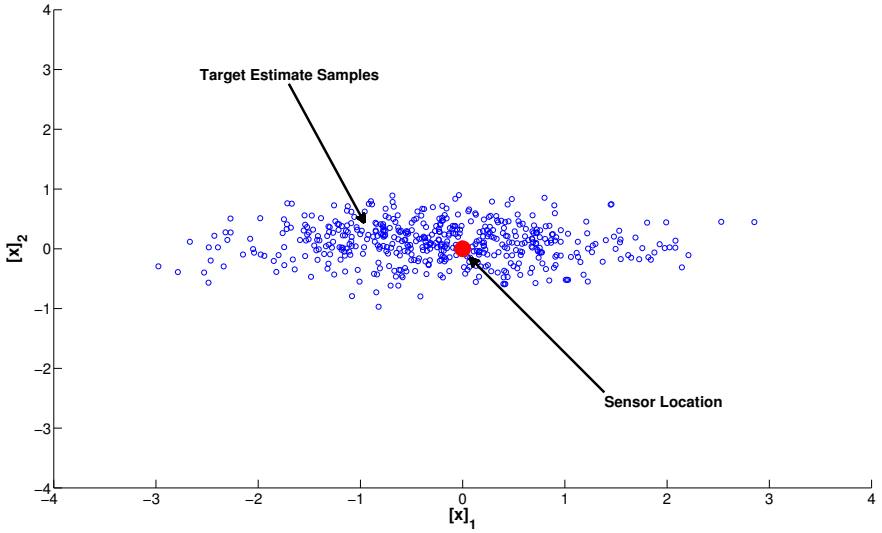


**Figure 6.1:** Sensor and Target-Estimate Scatter Plot for $[\sigma_1, \sigma_2] = [1, 0.1]$ and $\hat{\mathbf{l}} = [-0.2, 0.1]^\top$

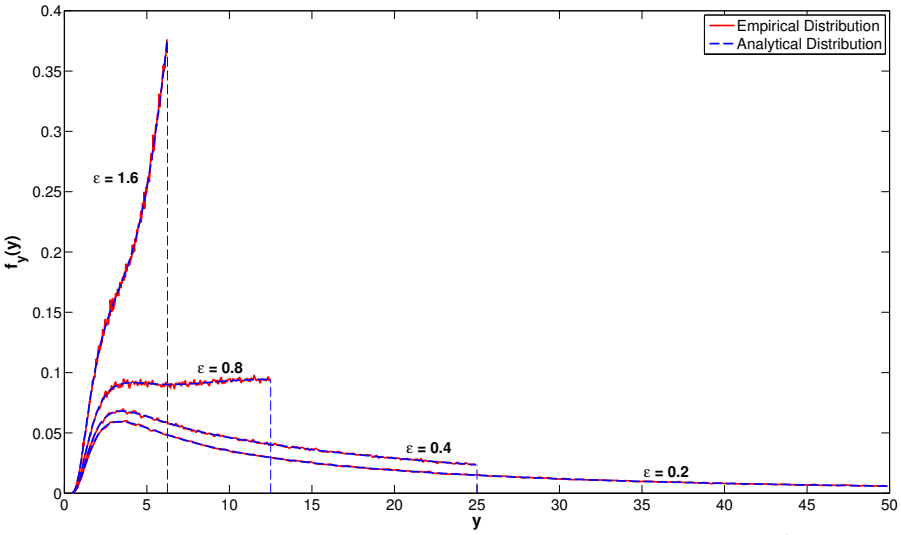


**Figure 6.2:** Measurement Distribution Plot with $[\sigma_1, \sigma_2] = [1, 0.1]$ and $\hat{\mathbf{l}} = [-0.2, 0.1]^\top$

Figure 6.2 shows the empirical and theoretical distributions for varying values of $\epsilon$ and with 200 Bessel function product terms computed in the summation of (6.13). Note how the support of each distribution is different; indeed this function is supported only on the interval $[0, \frac{\Phi}{\epsilon}]$. It is also evident that there is no discernible difference between the empirical and theoretical distributions; the approximation accuracy is high for these parameter values. A somewhat different situation is en-

countered when increasing the sensor-to-target-estimate distance (but leaving all other parameters the same as in the previous figures), as shown in Fig. 6.3.

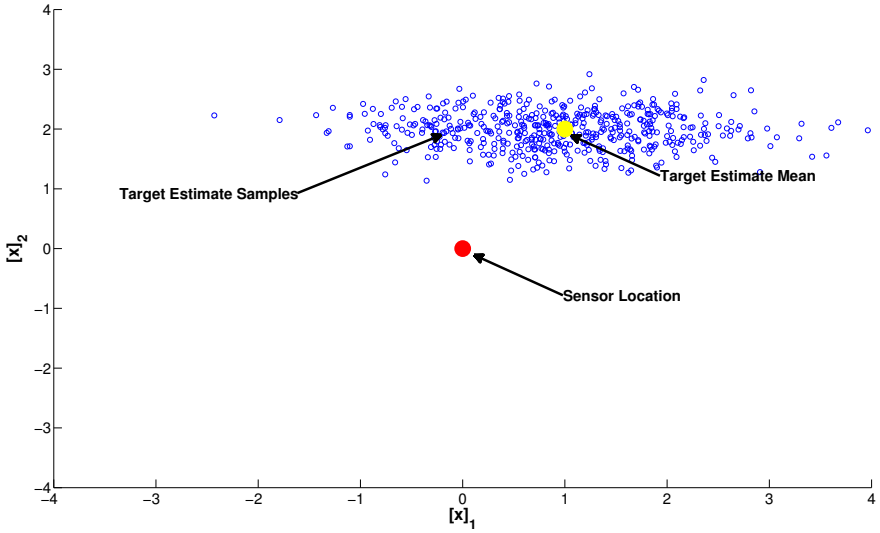


**Figure 6.3:** Sensor and Target-Estimate Scatter Plot for $[\sigma_1, \sigma_2] = [1, 0.1]$ and $\hat{\mathbf{l}} = [1, 2]^\top$

While (6.13) is well-approximated for larger values of $y_{\mathbf{l}}$, smaller values result in larger arguments for the Bessel series product terms causing a rapid deterioration in numerical accuracy. Taking more terms in the series does not alleviate this situation, as an unacceptable loss of numerical precision has already occurred. This problem is illustrated in Fig. 6.4.

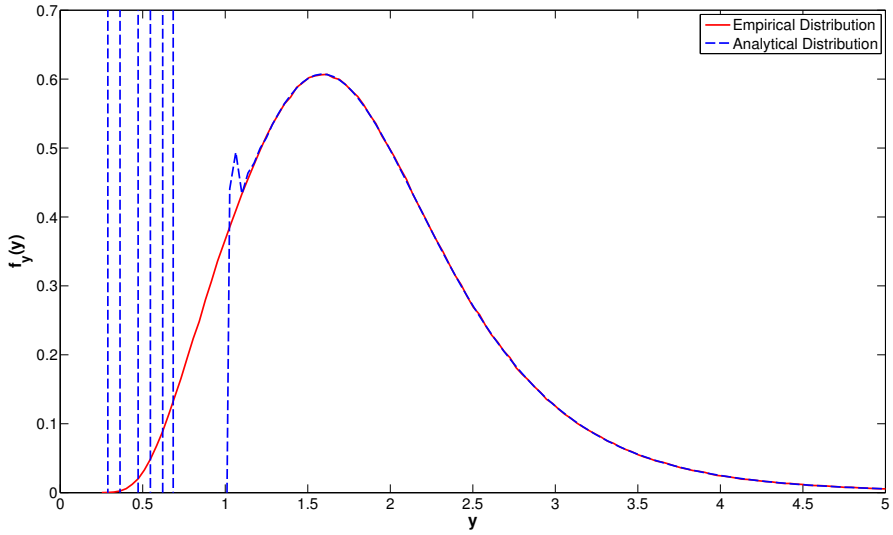


**Figure 6.4:** Measurement Distribution Plot using Solution 1 with $\epsilon = 0.1$,$[\sigma_1, \sigma_2] = [1, 0.1]$ and $\hat{\mathbf{l}} = [1, 2]^\top$

Attention is turned to the second method in order to combat these numerical issues. The sum is not computed directly (accumulating term by term) since this will again introduce difficulties

with numerical precision. Instead, for a chosen $N$ one can approximate (6.29) by evaluating the polynomial:

$$L\left[-\frac{\sigma_2^2-\sigma_1^2}{2\sigma_1^2\sigma_2^2}\eta_{y_k}^{\frac{2}{\alpha}}, \frac{\hat{d}_1}{\sigma_1^2}\eta_{y_k}^{\frac{1}{\alpha}}, \frac{\hat{d}_2}{\sigma_2^2}\eta_{y_k}^{\frac{1}{\alpha}}\right] \approx \sum_{u=0}^{N} a_u \left(\eta_{y_k}^{\frac{2}{\alpha}}\right)^u \tag{6.31}$$

where $a_u$ is calculated as:

$$a_u = \sum_{n=0}^{N}\sum_{k=0}^{n}\sum_{l=0}^{k}\left( \mathbb{1}\left[u=\frac{n+k-l}{2}\right]\frac{2\delta_{(n,k,l)}}{l!(n-k)!(k-l)!}\left(-\frac{\sigma_2^2-\sigma_1^2}{2\sigma_1^2\sigma_2^2}\right)^{k-l} \left(\frac{\hat{d}_1}{\sigma_1^2}\right)^{l}\left(\frac{\hat{d}_2}{\sigma_2^2}\right)^{n-k}\beta\left(\frac{n-k+1}{2},\frac{2k-l+1}{2}\right)\right) \tag{6.32}$$

In words, $a_u$ is the accumulation of all terms in the finite-sum approximation of (6.29) which have a common exponent for $\left(\eta_{y_k}^{\frac{2}{\alpha}}\right)^u$. Each of these coefficients can be computed in one pass, by accumulating individual coefficients separately within an array. Essentially, this method delays explicit computation of the polynomial until all coefficients have been fully aggregated, and dramatically helps to mitigate numerical errors that result from the loss of precision in adding numbers together with greatly varying orders of magnitude. Figure 6.5 shows equation (6.30) computed using $N = 130$ plotted alongside the empirical distribution, where it can be seen that the numerical issues have been adequately resolved.

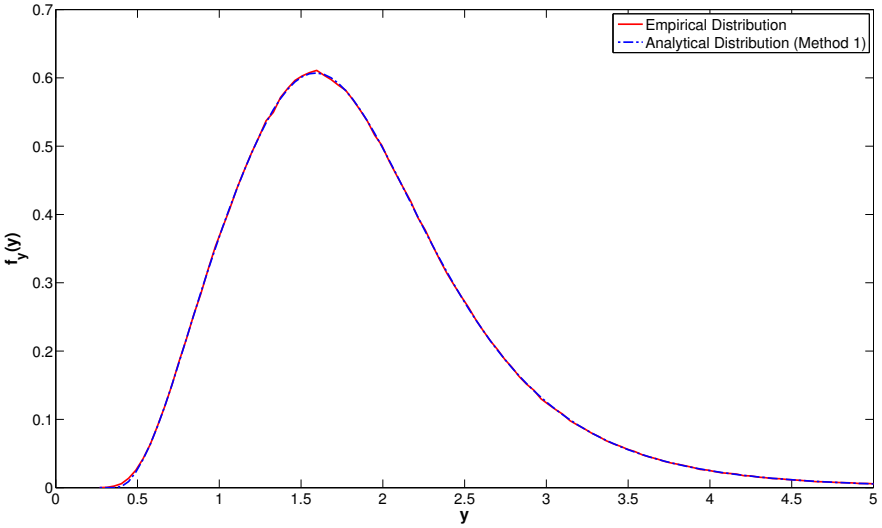


**Figure 6.5:** Measurement Distribution Plot using Method 1 with $\epsilon = 0.1$,$[\sigma_1, \sigma_2] = [1, 0.1]$ and $\hat{\mathbf{l}} = [1, 2]^\top$

### 6.1.2 Small Estimate Variances

As the target estimate variances are decreased, the situation becomes increasingly challenging; large values of $k_1$-$k_6$ push the values of the Bessel functions in the first method well into the large-argument region and make computation with sufficient accuracy intractable. The second method provides little immediate relief; the numerical range of the coefficients $a_u$ in (6.32) and the polynomial terms in (6.31) can be extremely large and quickly surpasses the typical 64-bit precision range of $\pm 10^{308}$. To be more precise, the large majority of the coefficients will have an order of magnitude *less* than this range while the polynomial terms may be larger than the maximum representable number; this poses the greatest numerical difficulty since each polynomial term/coefficient product may indeed be of sufficient order to affect the net sum in (6.31).

Nonetheless, with careful scaling (a "sliding" intermediate logarithmic weighting is used to ensure each product term remains within floating-point numerical range until the final sum/exponential multiplication in (6.30) is performed) it has been found that method 1 does indeed converge to sufficient accuracy for large enough $N$. An example of this case is shown in Fig. 6.6 with $\hat{\mathbf{l}} = \begin{bmatrix} -1.0 & -2.0 \end{bmatrix}^\top$ and $\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 0.01 \end{bmatrix}$. Notice the large inaccuracies present in moderate values for $N$ in Fig. 6.7. Nonetheless, convergence is indeed achieved for $N = 1400$ and accuracy is high, however this has come at a significant computational expense.

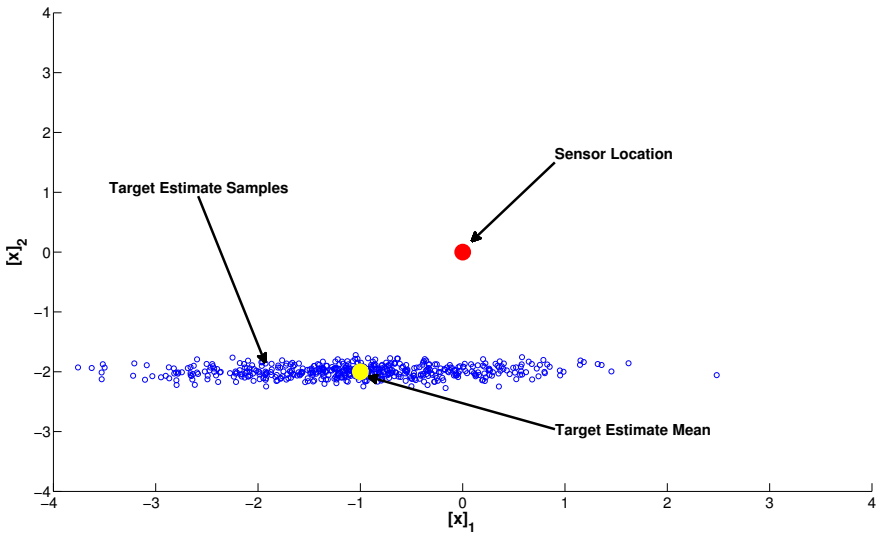


**Figure 6.6:** Sensor and Target-Estimate Scatter Plot for $\epsilon = 0.1$,$[\sigma_1, \sigma_2] = [1, 0.01]$ and $\hat{\mathbf{l}} = [-1, -2]^\top$

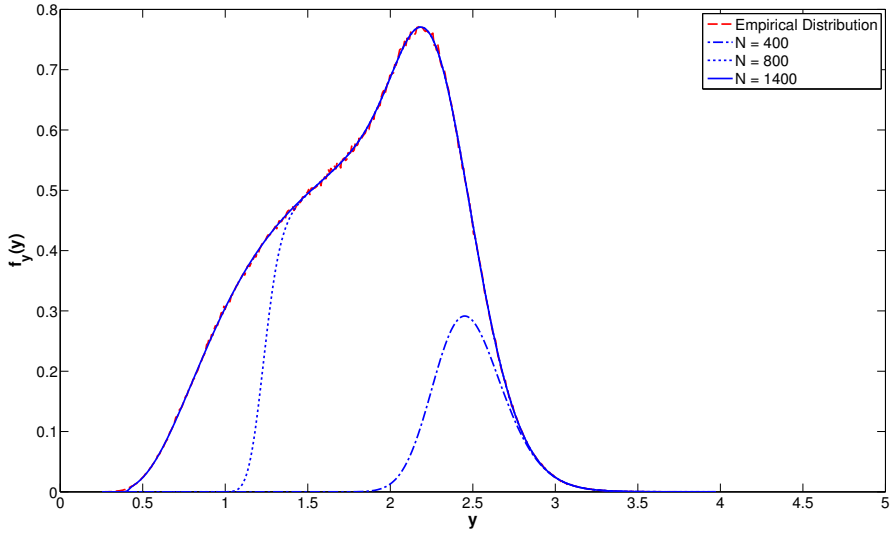


**Figure 6.7:** Measurement Distribution Plot using solution 2 with $\epsilon = 0.1$,$[\sigma_1, \sigma_2] = [1, 0.01]$ and $\hat{\mathbf{l}} = [-1, -2]^\top$ for various values of $N$

### 6.1.3 Analytical Approximations For A Single Sensor

#### 6.1.3.1 Diagonal Covariance

It is highly desirable to find reasonable approximations for (6.13) that will allow for practical computations. One possible approach involves restricting the component variances of $\mathbf{l}$ to be equal, i.e, $\sigma_1 = \sigma_2 = \sigma$. This results in a dramatic simplification of (6.13) by noting that then $k_5 = k_6 = 0$, and for $j \geq 0$, $\mathrm{I}_j(0) = 0$, and $\mathrm{I}_0(0) = 1$, yielding,

$$f\left(y_{\mathbf{l}} \mid \hat{\mathbf{l}}, \hat{\mathbf{C}}_t\right) = \frac{\Phi \eta_{y_{\mathbf{l}}}^{\left(\frac{2}{\alpha}-1\right)}}{\alpha \sigma^2 y_{\mathbf{l}}^2} e^{-\frac{1}{2\sigma^2}\left(\hat{d}_1^2 + \hat{d}_2^2 + \eta_{y_{\mathbf{l}}}^{\frac{2}{\alpha}}\right)}$$

$$\times \mathrm{I}_0\left(\frac{\|\hat{\mathbf{d}}\|}{\sigma^2} \eta_{y_{\mathbf{l}}}^{\frac{1}{\alpha}}\right). \tag{6.33}$$

When the constant factor $\frac{\|\hat{\mathbf{d}}\|}{\sigma_t^2}$ within the Bessel function argument in (6.33) is large, one can use the large argument approximation for Modified Bessel functions, $\mathrm{I}_0(x) \approx \frac{e^x}{\sqrt{2\pi x}}$. As a result, (6.33) can be approximated as

$$f\left(y_{\mathbf{l}} \mid \hat{\mathbf{l}}_t, \mathbf{C}_t\right) \approx \frac{\Phi \eta_{y_{\mathbf{l}}}^{\frac{3-2\alpha}{2\alpha}}}{\alpha \sigma \sqrt{2\pi}} \frac{e^{-\frac{1}{\sigma^2}\left(\|\hat{\mathbf{d}}\| - \eta_{y_{\mathbf{l}}}^{\frac{1}{\alpha}}\right)^2}}{y_{\mathbf{l}}^2 \sqrt{\|\hat{\mathbf{d}}\|}}, \tag{6.34}$$

which is valid on the domain of $f\left(y_{\mathbf{l}} \mid \hat{\mathbf{l}}, \hat{\mathbf{C}}\right)$, $\left[0 \leq y_{\mathbf{l}} \leq \frac{\Phi}{\epsilon}\right]$. With the given assumptions for the parameter values (particularly with small $\sigma$), the exponential in the numerator in (6.34) decays on both sides of $y_{\mathbf{l}}^* = \frac{\Phi}{\|\hat{\mathbf{d}}\|^\alpha + \epsilon}$ much faster than the remaining factors can change appreciably. We can

thus make the approximation that the remaining factors are constant with values obtained at $y_{\mathbf{l}}^*$. We now approximate the argument of the numerator exponential with the first two terms of its Taylor series expansion about $y_{\mathbf{l}}^*$. It can then be shown that $f_{y_{\mathbf{l}}}\left(y_{\mathbf{l}} \mid \hat{\mathbf{l}}, \hat{\mathbf{C}}\right) \approx \mathcal{N}\left(y_{\mathbf{l}}^*, \sigma^{*^2}\right)$ with,

$$\sigma^* = \frac{\Phi\alpha\sigma\|\mathbf{d}\|^{\alpha-1}}{\left(\|\hat{\mathbf{d}}\|^{\alpha}+\epsilon\right)^2}. \tag{6.35}$$

In Fig. 6.8, a comparison of the empirical histogram with the Gaussian approximation is presented for several values of $\sigma$, with the remaining parameters fixed at $\Phi = 10$, $\alpha = 1.6$, and $\hat{\mathbf{d}} = [3,4]^\top$.

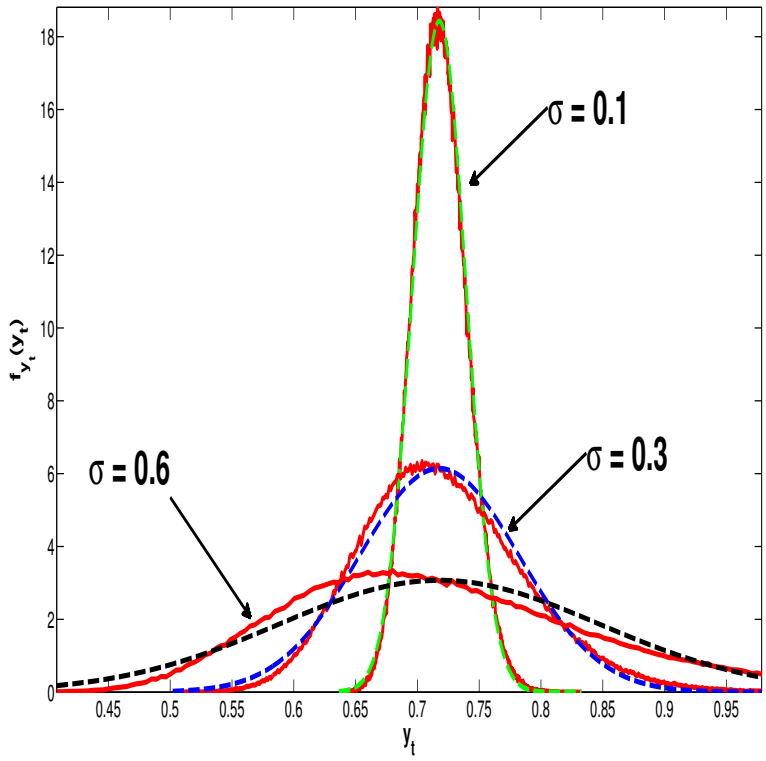


**Figure 6.8:** Gaussian approximation of $f\left(y_{\mathbf{l}} \mid \hat{\mathbf{l}}, \hat{\mathbf{C}}\right)$, where the solid line is the true density and the dashed line is the approximation of the density.

With this approximation in hand, we can return to the general multi-target scenario as expressed in (6.1) but still for only a single sensor. This expression can be further decomposed into the individual components contributing to the net measurement $y_{\mathbf{l}}$ as:

$$y_{\mathbf{l}} = h\left(\mathbf{x}\right) + \sum_{i=1}^{L} \varsigma_i + v_k \tag{6.36}$$

where again $h\left(\mathbf{x}\right)$ represents the target contribution that we aim to condition on, $v_k$ is the sensor noise, and $\varsigma_i$ represents the contribution to the measurement $y_{\mathbf{l}}$ from the $i$-th target. It must be

recognized that severe complications will arise in formulation of the actual likelihood function for this model. Each $\varsigma_i$ is alone distributed with form as in (6.13), but there is now an additional inter-dependency between each individual contribution since all measurement components are with respect to the common sensor location. However, as a first-order approximation, one can neglect this complication and assume each $\varsigma_i$ is independently distributed under the premise that each component is well approximated by a Gaussian with parameters defined around equation (6.35). With the sensor noise distributed as $\mathcal{N}\left(0, \sigma_v^2\right)$, one has that $y_{\mathbf{l}}$ is approximately distributed as $\mathcal{N}\left(\bar{y}_{\mathbf{l}}, \bar{\sigma}^2\right)$ with:

$$\begin{aligned} \bar{y}_k &= h\left(\mathbf{x}\right) + \sum_{i=1}^{L} \frac{\Phi}{\|\hat{\mathbf{d}}_i\|^{\alpha} + \epsilon} \\ \bar{\sigma} &= \sqrt{\alpha^2 \Phi^2 \sum_{i=1}^{L} \frac{\max\left(\sigma_{l,1}, \sigma_{l,2}\right)^2 \|\mathbf{d}\|^{2(\alpha-1)}}{\left(\|\mathbf{d}\|^{\alpha} + \epsilon\right)^4} + \sigma_v^2} \end{aligned} \tag{6.37}$$

To investigate the performance of this approximation, let us take a scenario with $\Phi = 10$, $\epsilon = 0.1$, $h\left(\mathbf{x}\right) = 1$, $\sigma_v = 0.01$ and with 6 foreign target estimate means uniformly distributed (with a small random jitter applied) about a circle of radius $r$ centered at the sensor location. Figure 6.9 shows a scatter plot of the case where $r = 3$ and the target estimate variances are ideally small. A comparison of the empirical distribution and Gaussian approximation for this case is shown in Fig. 6.10 where it can be seen that the approximation is reasonably accurate.

To illustrate the limitations of using this approximation, let us examine two cases where the parameters begin to deviate from the original assumptions. Figures 6.11 and 6.12 show the same case except where $\sigma_l$ for all targets is increased by a factor of 5. Here it can be seen that the empirical distribution has departed from the approximation in the form of a skew to the right; the distribution decays more rapidly than a Gaussian to the left and is heavier-tailed on the right side.

The third case shows a more severe departure from this approximation. In this case the variances are returned to that of the first example, but the radius about the sensor location is decreased to $r = 1.0$. Figure 6.13 shows the scatter plot for this case and Fig. 6.14 shows the corresponding distribution. Notice in this specific realization there is one target estimate mean that is located very close to the sensor (specifically at a distance of 0.2 units); this represents a large deviation from the assumption of moderate sensor-target distance and has a significant impact on the likelihood function.

It has been shown that potentially severe deterioration of the Gaussian approximation in (6.37) will occur with either large target-estimate variances, or nearly co-located sensor/target configurations. While these situations represent abnormal conditions and must be handled differently, they are not typical of the tracking environment that is considered within MASCOT. Prior to cooperative tracking, each filter is assumed to have well established tracking for its own target (small

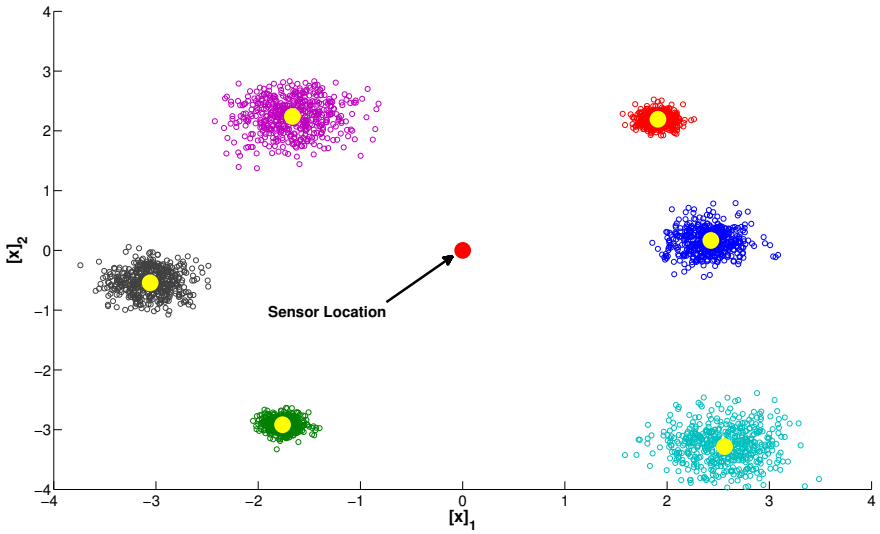


**Figure 6.9:** Sensor and 6 Target-Estimate Scatter Plot for $\epsilon = 0.1$, $\sigma = [0.2, 0.1, 0.3, 0.2, 0.1, 0.3]$ and $r = 3$

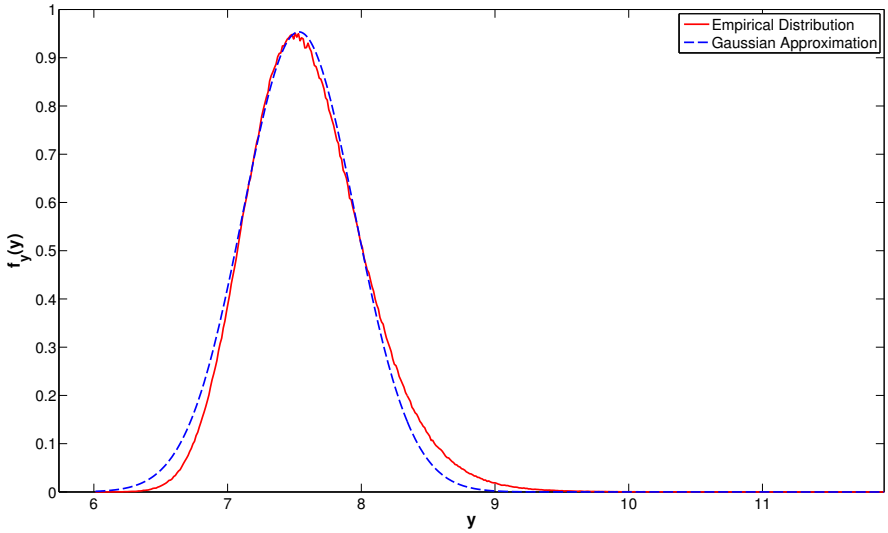


**Figure 6.10:** Sensor and 6 Target-Estimate Distribution Plot for $\epsilon = 0.1$, $\sigma = [0.2, 0.1, 0.3, 0.2, 0.1, 0.3]$ and $r = 3$

estimate covariance matrix). As the sensors are mobile, they are designed to maintain a fixed distance from their own target while maintaining as large a distance as possible from others so as to avoid excessive interference in the measurements. In this case, the measurement component distributions are reasonably approximated by (6.35) and allow (6.37) to provide a fairly accurate and practical approximation of the multi-target likelihood function for a single sensor.

#### 6.1.3.2 General Covariance Matrix

The situation is significantly more complicated if the full generality of $\hat{\mathbf{C}}$ is maintained. Rather than attempting to develop an approximation directly from (6.13), let us return to the expression

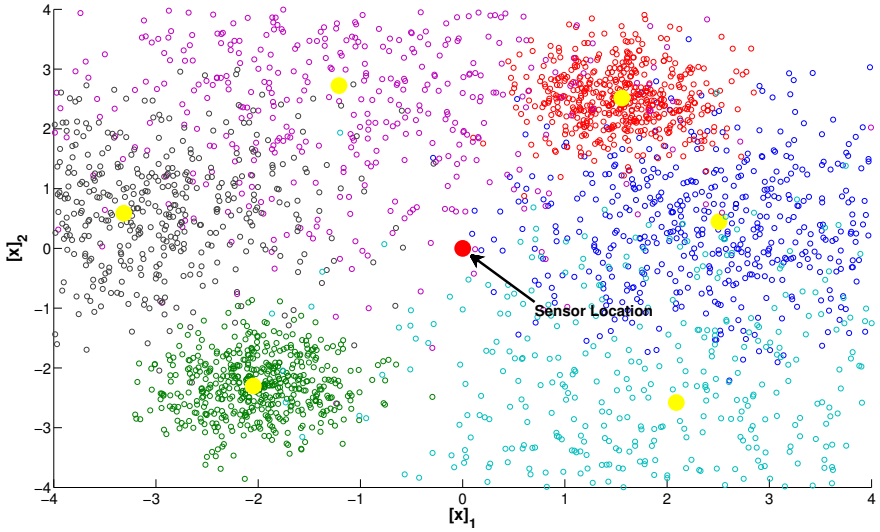


**Figure 6.11:** Sensor and 6 Target-Estimate Scatter Plot for $\epsilon = 0.1$, $\sigma = [1.0, 0.5, 1.5, 1.0, 0.5, 1.5]$ and $r = 3$

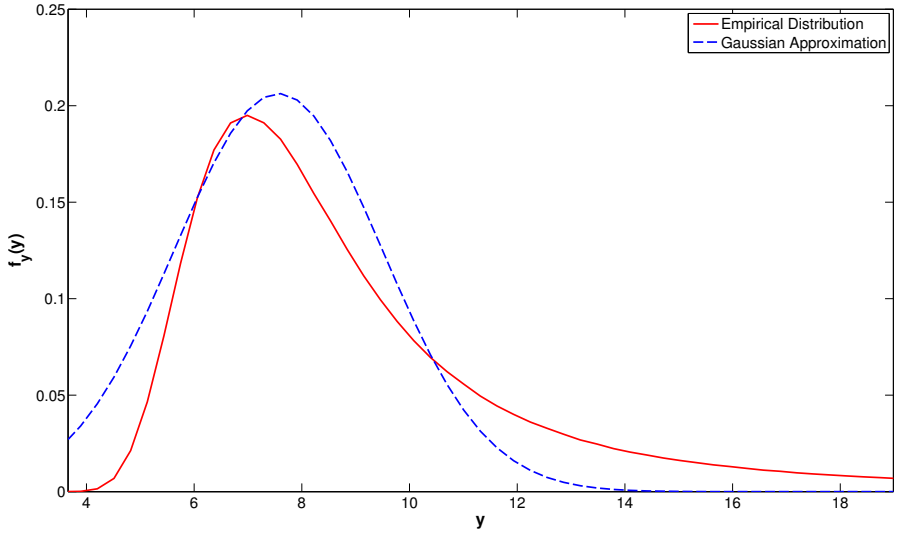


**Figure 6.12:** Sensor and 6 Target-Estimate Distribution Plot for $\epsilon = 0.1$, $\sigma = [1.0, 0.5, 1.5, 1.0, 0.5, 1.5]$ and $r = 3$

for the joint density in (6.8). Recall that we must compute the integral,

$$\int_0^{2\pi} f_{y_1,\theta}\left(y_1, \theta \mid \hat{\mathbf{l}}, \mathbf{C}\right) d\theta = \left[\frac{\Phi \eta_{y_1}^{\left(\frac{2}{\alpha}-1\right)}}{2\pi\alpha\sigma_1\sigma_2 y_1^2\sqrt{1-\rho^2}}\right] \int_0^{2\pi} \exp\left[\mathfrak{I}_1\left(\eta_{y_1}, \theta, \hat{\mathbf{d}}, \hat{\mathbf{C}}\right)\right] d\theta,$$

to obtain $f\left(y_1 \mid \hat{\mathbf{l}}, \mathbf{C}\right)$. Integrals of this form can be approximated analytically through application of the method of Laplace, i.e.,

$$\int_0^{2\pi} e^{\lambda g(\theta)} d\theta \approx \left(\sqrt{\frac{2\pi}{\lambda|\frac{\partial^2 g(\theta)}{\partial\theta^2}|_{\theta=\theta^*}}}\right) e^{\lambda g(\theta^*)} \quad \text{as } \lambda \to \infty, \tag{6.38}$$

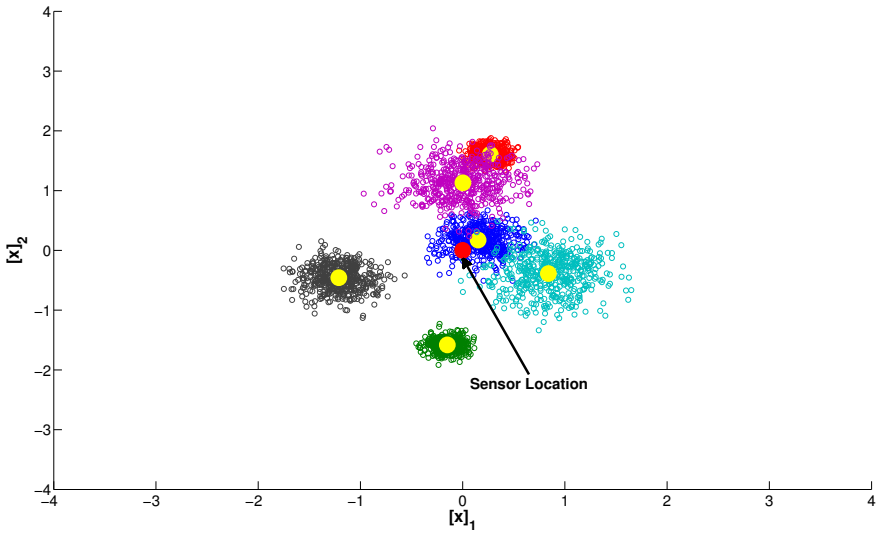


**Figure 6.13:** Sensor and 6 Target-Estimate Scatter Plot for $\epsilon = 0.1$, $\sigma = [0.2, 0.1, 0.3, 0.2, 0.1, 0.3]$ and $r = 1$

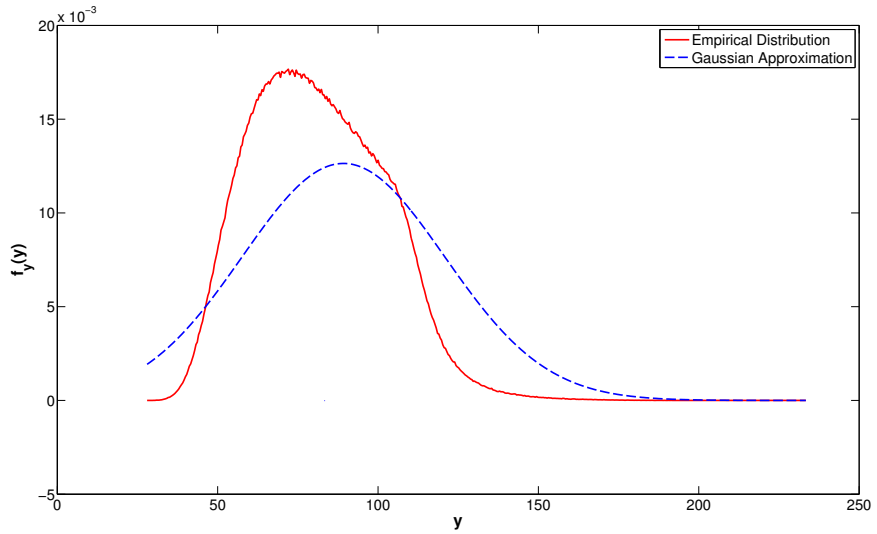


**Figure 6.14:** Sensor and 6 Target-Estimate Distribution Plot for $\epsilon = 0.1$, $\sigma = [0.2, 0.1, 0.3, 0.2, 0.1, 0.3]$ and $r = 1$

where

$$\theta^* = \arg\max_{\theta} g(\theta).$$

To clarify how this can be applied to (6.38), using again the parameters in (6.11), we can write,

$$\mathcal{T}_1\left(\eta_{y_1}, \theta, \hat{\mathbf{d}}, \mathbf{C}\right) = \mathcal{T}_2\left(\eta_{y_1}, \hat{\mathbf{d}}, \mathbf{C}\right) + \mathcal{T}_3\left(\eta_{y_1}, \theta, \hat{\mathbf{d}}, \mathbf{C}\right), \tag{6.39}$$

where

$$\mathcal{T}_2\left(\eta_{y_1}, \hat{\mathbf{d}}, \mathbf{C}\right) = -\frac{1}{2(1-\rho^2)}\left(k_1 + \left(k_2 - \sqrt{k_5^2 + k_6^2}\right)\eta_{y_1}^{\frac{2}{\alpha}}\right), \tag{6.40}$$

$$\mathcal{T}_3\left(\eta_{y_1},\theta,\hat{\mathbf{d}},\mathbf{C}\right)=-\frac{1}{2(1-\rho^2)}\left(\left(\sqrt{k_3^2+k_4^2}\right)\eta_{y_1}^{\frac{1}{\alpha}}\cos\left(\theta-\phi_1\right)+2\left(\sqrt{k_5^2+k_6^2}\right)\eta_{y_1}^{\frac{2}{\alpha}}\cos^2\left(\theta-\phi_2\right)\right). \tag{6.41}$$

We can then form the approximation,

$$f\left(y_1\mid\hat{\mathbf{l}},\mathbf{C}\right)\approx\left[\frac{\Phi\eta_{y_1}^{\left(\frac{2}{\alpha}-1\right)}}{2\pi\alpha\sigma_1\sigma_2y_1^2\sqrt{1-\rho^2}}\right]\left(\sqrt{\frac{2\pi}{\left|\frac{\partial^2\mathcal{T}_3\left(\eta_{y_1},\theta,\hat{\mathbf{d}},\hat{\mathbf{C}}\right)}{\partial\theta^2}\right|_{\theta=\theta^*}}}\right)$$
$$\times\exp\left[\mathcal{T}_2\left(\eta_{y_1},\hat{\mathbf{d}},\mathbf{C}\right)+\mathcal{T}_3\left(\eta_{y_1},\theta^*,\hat{\mathbf{d}},\mathbf{C}\right)\right] \tag{6.42}$$

Note that it is somewhat more appropriate to express (6.41) as

$$\mathcal{T}_3\left(\eta_{y_1},\theta,\hat{\mathbf{d}},\mathbf{C}\right)=\lambda_1\mathcal{G}_1\left(\eta_{y_1},\theta,\hat{\mathbf{d}},\mathbf{C}\right)+\lambda_2\mathcal{G}_2\left(\eta_{y_1},\theta,\hat{\mathbf{d}}_t,\mathbf{C}\right), \tag{6.43}$$

where

$$\mathcal{G}_1\left(\eta_{y_1},\theta,\hat{\mathbf{d}},\mathbf{C}\right)=\cos(\theta-\phi_1), \tag{6.44}$$
$$\mathcal{G}_2\left(\eta_{y_1},\theta,\hat{\mathbf{d}},\mathbf{C}\right)=\cos^2(\theta-\phi_2), \tag{6.45}$$

and

$$\lambda_1=\frac{-\sqrt{k_3^2+k_4^2}}{2(1-\rho^2)}\eta_{y_1}^{\frac{1}{\alpha}},\quad\lambda_2=\frac{-\sqrt{k_5^2+k_6^2}}{(1-\rho^2)}\eta_{y_1}^{\frac{2}{\alpha}}. \tag{6.46}$$

We observe that we must have both $\lambda_1$ and $\lambda_2$ large for reasonable approximation accuracy. A complication can occur when the two constants have the same order of magnitude that can significantly degrade the accuracy of (6.42), in which case, an extended version of the Laplace Method, as in [89], is needed. Nonetheless, it has been found empirically that for most parameter values of interest (with sufficiently small $\sigma_1$ and $\sigma_2$), (6.42) does provide excellent accuracy.

Note that an explicit expression for $\theta^*$ has not been given here. While an exact analytical form does exist, it is considerably complicated and prone to numerical instability. We can instead choose to apply a basic numerical approach (which has been found to work well in this setting), or seek an approximation to $\theta^*$ that will not compromise the accuracy of (6.42). In fact, by pursuing the latter method, we can develop a significantly simpler form for approximation of $f_{y_1}\left(y_1\mid\hat{\mathbf{l}},\mathbf{C}\right)$, which is now described in detail.

We can begin by noting that $\theta^*$ must satisfy the condition,

$$\left[\frac{\partial}{\partial\theta}\mathcal{T}_1\left(\eta_{y_1},\theta,\hat{\mathbf{d}},\mathbf{C}\right)\right]_{\theta=\theta^*}=0. \tag{6.47}$$

Recall that $\mathcal{T}_1\left(\eta_{y_\mathbf{l}}, \theta, \hat{\mathbf{d}}, \mathbf{C}\right)$ represents the argument of the exponential in $\mathcal{N}\left(\eta_{y_\mathbf{l}}^{\frac{1}{\alpha}} \begin{bmatrix}\cos\theta \\ \sin\theta\end{bmatrix} \,\middle|\, \hat{\mathbf{d}}, \mathbf{C}\right)$ and can be expressed as,

$$\mathcal{T}_1\left(\eta_{y_\mathbf{l}}, \theta, \hat{\mathbf{d}}, \mathbf{C}\right) = -\frac{1}{2(1-\rho^2)}\left[\frac{1}{\sigma_1^2}\left(\eta_{y_\mathbf{l}}^{\frac{1}{\alpha}}\cos\theta - \hat{d}_1\right)^2 + \frac{1}{\sigma_2^2}\left(\eta_{y_\mathbf{l}}^{\frac{1}{\alpha}}\sin\theta - \hat{d}_2\right)^2 - \frac{2\rho}{\sigma_1\sigma_2}\left(\eta_{y_\mathbf{l}}^{\frac{1}{\alpha}}\cos\theta - \hat{d}_1\right)\left(\eta_{y_\mathbf{l}}^{\frac{1}{\alpha}}\sin\theta - \hat{d}_2\right)\right], \quad (6.48)$$

from which we obtain by the condition in (6.47),

$$\left(\frac{\rho}{\sigma_1\sigma_2}\cos\theta^* + \frac{1}{\sigma_1^2}\sin\theta^*\right)\left(\eta_{y_\mathbf{l}}^{\frac{1}{\alpha}}\cos\theta^* - \hat{d}_1\right) = \left(\frac{\rho}{\sigma_1\sigma_2}\sin\theta^* + \frac{1}{\sigma_2^2}\cos\theta^*\right)\left(\eta_{y_\mathbf{l}}^{\frac{1}{\alpha}}\sin\theta^* - \hat{d}_2\right). \quad (6.49)$$

This can be rewritten as a condition on $\eta_{y_\mathbf{l}}$ that is satisfied when $\theta^*$ is indeed a critical point,

$$\eta_{y_\mathbf{l}}^{\frac{1}{\alpha}} = h(\theta^*) = \frac{\left[\left(\frac{\rho\hat{d}_1}{\sigma_1\sigma_2} - \frac{\hat{d}_2}{\sigma_2^2}\right)\cos\theta^* + \left(\frac{\hat{d}_1}{\sigma_1^2} - \frac{\rho_t\hat{d}_2}{\sigma_1\sigma_2}\right)\sin\theta^*\right]}{\left[\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right)\sin\theta^*\cos\theta^* + \left(\frac{\rho}{\sigma_1\sigma_2}\right)\cos 2\theta^*\right]}. \quad (6.50)$$

Equipped with (6.50), we can implicitly construct a first-order Taylor series approximation to $\theta^*$,

$$\theta^* \approx \theta_0^* + \left[\frac{\partial h(\theta^*)}{\partial\theta}\right]^{-1}_{\theta^*=\theta_0^*}\left(\eta_{y_\mathbf{l}}^{\frac{1}{\alpha}} - h(\theta_0^*)\right), \quad (6.51)$$

where $\theta_0^*$ represents the point about which this expansion is taken. This point is here chosen to coincide with the value of $h(\theta_0^*)$ that will minimize $\mathcal{T}_1\left(h(\theta_0^*), \theta_0^*, \hat{\mathbf{d}}, \mathbf{C}\right)$. It can be shown that this yields the maximum of $f\left(y_\mathbf{l} \mid \hat{\mathbf{l}}, \mathbf{C}\right)$ and its value is,

$$\theta_0^* = \arccos\left(\frac{\hat{d}_1}{\|\hat{\mathbf{d}}\|}\right) \quad \text{at} \quad h(\theta_0^*) = \sqrt{\hat{d}_1^2 + \hat{d}_2^2}, \quad (6.52)$$

Towards finding a compact analytical approximation to $f_{y_\mathbf{l}}\left(y_\mathbf{l} \mid \hat{\mathbf{l}}, \mathbf{C}\right)$, let us also make use of the first-order Taylor series expansions about $\theta_0^*$ for all the trigonometric terms in (6.48), i.e.,

$$\begin{aligned}\cos\theta &\approx \cos\theta_0^* - \sin\theta_0^*\left(\theta - \theta_0^*\right), \\ \sin\theta &\approx \sin\theta_0^* + \cos\theta_0^*\left(\theta - \theta_0^*\right).\end{aligned} \quad (6.53)$$

By following similar steps as in the preceding development of the approximation given in (6.42), except in this case retaining the original form of $\mathcal{T}_1\left(\eta_{y_\mathbf{l}}, \theta, \hat{\mathbf{d}}, \mathbf{C}\right)$, we arrive at,

$$f_{y_\mathbf{l}}\left(y_\mathbf{l} \mid \hat{\mathbf{l}}, \mathbf{C}\right) \approx \left[\frac{\Phi\eta_{y_\mathbf{l}}^{\left(\frac{2}{\alpha}-1\right)}}{2\pi\alpha\sigma_1\sigma_2 y_\mathbf{l}^2\sqrt{1-\rho^2}}\right]\left(\sqrt{\frac{2\pi}{\left|\frac{\partial^2\mathcal{T}_1\left(\eta_{y_\mathbf{l}}, \theta, \hat{\mathbf{d}}, \mathbf{C}\right)}{\partial\theta^2}\right|_{\theta=\theta^*}}}\right)\exp\left[\mathcal{T}_1\left(\eta_{y_\mathbf{l}}, \theta^*, \hat{\mathbf{d}}, \mathbf{C}\right)\right]. \quad (6.54)$$

The exponential term in (6.54) can be greatly simplified by substituting (6.52) and (6.53) into (6.48) yielding

$$\exp\left[\mathcal{T}_1\left(\eta_{y_\mathbf{l}},\theta^*,\hat{\mathbf{d}},\mathbf{C}\right)\right]\approx\exp\left[-\mathcal{K}\left(\|\hat{\mathbf{d}}\|-\eta_{y_\mathbf{l}}^{\frac{1}{\alpha}}\right)^2\right] \tag{6.55}$$

$$\mathcal{K}=\frac{\left(\hat{d}_1^2+\hat{d}_2^2\right)}{2\left(\hat{d}_1^2\sigma_1^2+2\hat{d}_1\hat{d}_2\rho_t\sigma_1\sigma_2+\hat{d}_2^2\sigma_2^2\right)}=\frac{\|\hat{\mathbf{d}}\|^2}{2\hat{\mathbf{d}}^\top\mathbf{C}\hat{\mathbf{d}}} \tag{6.56}$$

As long as $\mathcal{K}$ is relatively large, this term will rapidly decay to zero for values of $\eta_{y_\mathbf{l}}^{\frac{1}{\alpha}}$ away from $\|\hat{\mathbf{d}}\|$ allowing us to approximate the remaining terms in (6.54) that multiply the exponential, as a constant evaluated at $\eta_{y_\mathbf{l}}^{\frac{1}{\alpha}}=h\left(\theta_0^*\right)$ and $\theta^*=\theta_0^*$. Proceeding in this fashion, we can show that the expression in (6.54) can be reduced by applying two more approximations,

$$\begin{aligned}\left|\frac{\partial^2\mathcal{T}_1\left(\eta_{y_\mathbf{l}},\theta,\hat{\mathbf{d}},\mathbf{C}\right)}{\partial\theta^2}\right|_{\theta=\theta^*}&\approx\frac{\hat{d}_1^2\sigma_1^2+2\hat{d}_1\hat{d}_2\rho\sigma_1\sigma_2+\hat{d}_2^2\sigma_2^2}{\left(1-\rho^2\right)\sigma_1^2\sigma_2^2}\\ \frac{\Phi\eta_{y_\mathbf{l}}^{\left(\frac{2}{\alpha}-1\right)}}{2\pi\alpha\sigma_1\sigma_2y_\mathbf{l}^2\sqrt{1-\rho^2}}&\approx\frac{\left(\|\hat{\mathbf{d}}\|^\alpha+\epsilon\right)^2}{\sqrt{2\pi}\alpha\Phi\|\hat{\mathbf{d}}\|^{(\alpha-1)}}.\end{aligned} \tag{6.57}$$

Allowing the following approximation,

$$f\left(y_\mathbf{l}\mid\hat{\mathbf{l}},\mathbf{C}\right)\approx\frac{\left(\|\hat{\mathbf{d}}^\alpha\|+\epsilon\right)^2\sqrt{\mathcal{K}}}{\sqrt{\pi}\alpha\Phi\|\hat{\mathbf{d}}\|^{(\alpha-1)}}\times\exp\left[-\mathcal{K}\left(\|\hat{\mathbf{d}}\|-\eta_{y_\mathbf{l}}^{\frac{1}{\alpha}}\right)^2\right]. \tag{6.58}$$

This result is similar to the one in (6.34) developed for the case where $\sigma_1=\sigma_2$. By following the same argument as in that situation, we have the generalized Gaussian approximation given by

$$f\left(y_\mathbf{l}\mid\hat{\mathbf{l}},\mathbf{C}\right)\approx\mathcal{N}\left(\hat{y}_k,\hat{\sigma}^2\right), \tag{6.59}$$

with,

$$\hat{y}_\mathbf{l}=\frac{\Phi}{\|\hat{\mathbf{d}}\|^\alpha+\epsilon}\qquad\hat{\sigma}_\mathbf{l}=\frac{\Phi\alpha\|\hat{\mathbf{d}}\|^{(\alpha-1)}}{\sqrt{2\mathcal{K}}\left(\|\hat{\mathbf{d}}\|^\alpha+\epsilon\right)^2}. \tag{6.60}$$

Equipped with the exact form (6.13) along with its subsequent approximation (6.59), we can form the general expression for the measurement likelihood as

$$f\left(y_k\mid\mathbf{x},\hat{\mathbf{l}},\mathbf{C}\right)=f\left(y_\mathbf{l}-h\left(\mathbf{x}\right)\mid\mathbf{x},\hat{\mathbf{l}},\mathbf{C}\right)*f\left(v_k\right), \tag{6.61}$$

where $*$ represents a convolution of densities and it is understood that $h\left(\mathbf{x}\right)$ is not random here due to conditioning. When the Gaussian approximation given in (6.59) is used and assuming $v_k\sim\mathcal{N}\left(0,\sigma_v^2\right)$, the measurement likelihood simplifies to $\mathcal{N}\left(y_k\mid h\left(\mathbf{x}\right)+\hat{y}_\mathbf{l},\hat{\sigma}^2+\sigma_v^2\right)$. If a non-Gaussian form such as (6.58) is used, one can always perform numerical integration to compute the convolution, which has been found to perform reasonably well when $v_k$ is Gaussian.

Returning to the original model in (6.1) where there exist $L$ interferers, one can use the Gaussian approximation presented to form the desired expression as[1],

$$f\left(y_k \mid h\left((\mathbf{x}\right), \hat{\mathbf{l}}_{1:L}, \mathbf{C}_{1:L}\right) \approx \mathcal{N}\left(y_k \mid h\left(\mathbf{x}\right) + \tilde{y}, \tilde{\sigma}^2\right) \tag{6.62}$$

where,

$$\tilde{y} = \sum_{i=1}^{L} \hat{y}_{\mathbf{l}_i} \quad \tilde{\sigma}^2 = \sum_{i=1}^{L} \hat{\sigma}^2_{\mathbf{l}_i} + \sigma_v^2 \tag{6.63}$$

where the individual terms $(\hat{y}_{\mathbf{l}_i}, \hat{\sigma}_{\mathbf{l}_i})$ are computed for the $i$-th interferer using equation (6.60). This approximation lends itself well to efficient computation and is of high value in performing agent cooperation within MASCOT.

## 6.2 Multiple Sensors

Considering a more generalized version of the model described involving multiple sensors reveals some interesting results. It is now desired to establish a form for the joint measurement likelihood $f\left(\mathbf{y}|\mathbf{x}, \hat{\mathbf{l}}_{1:L}, \mathbf{C}_{1:L}\right)$. Here it is understood that $\mathbf{x}$ represents the location of the target being tracked by the agent, $\hat{\mathbf{l}}_{1:L}$ represent interference location estimates communicated by other agents in the tracking environment, and $\mathbf{C}_{1:L}$ represent the respective covariance matrices of these communicated estimates.

The standard bootstrap particle filter algorithm assumes the joint likelihood can be factored as $\prod_{k=1}^{K} f\left(y_k|\mathbf{x}, \hat{\mathbf{l}}_{1:L}, \mathbf{C}_{1:L}\right)$, which is valid since the measurements are indeed independent when conditioned on the actual target location (or a particle representation of it) and with assumed independent sensor noise components. Such independence does not hold in this case; each measurement is not only dependent on $\mathbf{x}$, but is also a nonlinear function of the unknown random locations of the interferers $\mathbf{l}_{1:L}$. As such, there will always be some level of correlation between individual sensor measurements for a given agent. The actual nature of this correlation is strongly influenced by several factors, including the total number of targets in the environment and the current "spatial configuration" of all sensors and targets.

To further illustrate this concept, let us consider the relatively simple case of two sensors tracking one target $\mathbf{x}$, and receiving information from another agent regarding a single interferer $\mathbf{l}$, within the environment. The measurement model for this scenario is as follows:

$$\begin{aligned} y_1 &= \frac{\Phi}{\|\mathbf{q}_1 - \mathbf{x}\|^\alpha + \epsilon} + \frac{\Phi}{\|\mathbf{q}_1 - \mathbf{l}\|^\alpha + \epsilon} + v_1 \\ y_2 &= \frac{\Phi}{\|\mathbf{q}_2 - \mathbf{x}\|^\alpha + \epsilon} + \frac{\Phi}{\|\mathbf{q}_2 - \mathbf{l}\|^\alpha + \epsilon} + v_2 \end{aligned} \tag{6.64}$$

[1] Assuming the approximation is valid for each interferer. Otherwise one can always perform repeated convolution.

where $\mathbf{q}_k = \begin{bmatrix} q_k^1 & q_k^2 \end{bmatrix}$, and $v_k$ are again the respective location, and noise component of the $k$-th sensor. The interferer is assumed randomly distributed as $\mathbf{l} \sim \mathcal{N}\left(\hat{\mathbf{l}}, \mathbf{C}\right)$, where $\bar{\mathbf{x}}_f$ and $\mathbf{C}$. For the time being it is also assumed there is no sensor noise present ($v_k = 0 \, \forall k$).

Formulation of the joint likelihood $f(\mathbf{y}|\mathbf{x})$ proceeds by rewriting (6.64) as:

$$\begin{aligned} (q_1^1 - l_1)^2 + (q_1^2 - l_2)^2 &= \left( \frac{\Phi}{y_1 - h(\mathbf{q}_1, \mathbf{x})} - \epsilon \right)^{\frac{2}{\alpha}} \\ (q_2^1 - l_1)^2 + (q_2^2 - l_2)^2 &= \left( \frac{\Phi}{y_2 - h(\mathbf{q}_2, \mathbf{x}) - \epsilon} \right)^{\frac{2}{\alpha}} \end{aligned} \tag{6.65}$$

where $h(\mathbf{q}_k, \mathbf{x})$ is the contribution from $\mathbf{x}$ to the $k$-th sensor measurement. If we view (6.65) in terms of the two variables $l_1$ and $l_2$ with $\mathbf{y}$ fixed, then we can immediately recognize that these equations describe two circles; one centered at $(q_1^1, q_1^2)$ and the other at $(q_2^1, q_2^2)$ with radii:

$$r_1 = \left( \frac{\Phi}{y_1 - h(\mathbf{q}_1, \mathbf{x})} - \epsilon \right)^{\frac{1}{\alpha}} \qquad r_2 = \left( \frac{\Phi}{y_2 - h(\mathbf{q}_2, \mathbf{x})} - \epsilon \right)^{\frac{1}{\alpha}} \tag{6.66}$$

For given values of $r_1$ and $r_2$, the interferer location $\mathbf{l}$ must coincide with one of two possible intersection points of these circles. If the circles do not intersect, then there is no value for $\mathbf{l}$ which could have generated $\mathbf{y}$ and hence, in the absence of sensor noise, this measurement vector is impossible.

Given the vector $\mathbf{y}$ which generates the radius pair $(r_1, r_2)$, one can solve for the set of intersection points $\mathbf{l}^\star[r_1, r_2]$ (which may be empty) by referring to Fig. 6.15.

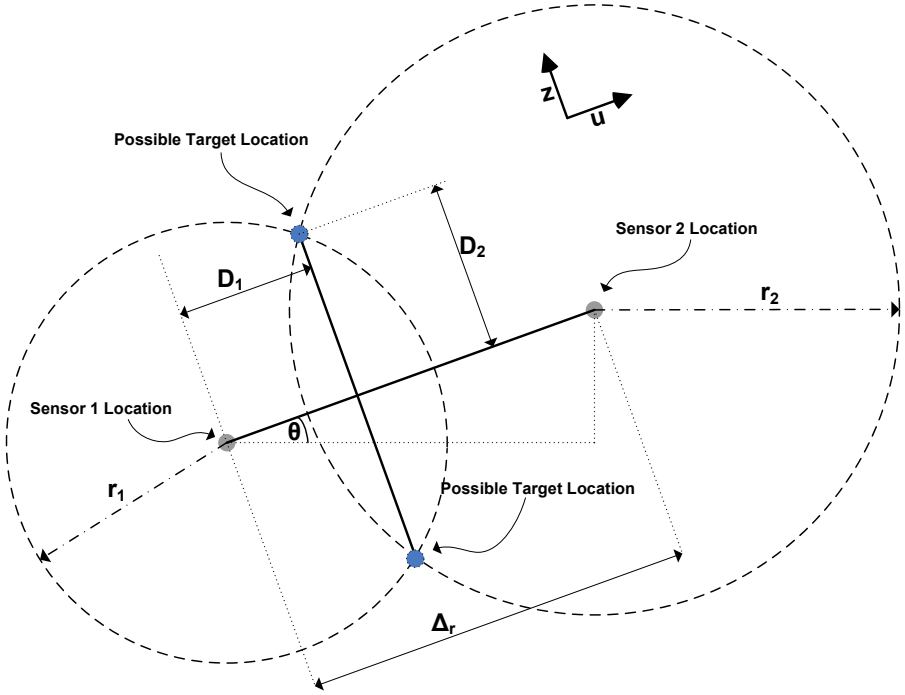


**Figure 6.15:** Two Sensor, Single Target Measurement Geometry

Note the coordinate system comprised of the auxiliary variables $(u, v)$ whose origin has been translated from $(0, 0)$ to coincide with $\mathbf{q}_1$ and then rotated counter-clockwise by $\theta$ to align with the line segment connecting the two circle centers. The length of this line is denoted $\Delta_r$ and is equal to $\sqrt{\left(q_1^1 - q_2^1\right)^2 + \left(q_1^2 - q_2^2\right)^2}$. One can find the distance from the center of the first circle, labeled $D_1$ in the diagram, along $\Delta_r$ to the perpendicular line intersecting the points in $\mathbf{l}^\star[r_1, r_2]$; this is in fact the radical line of the two circles. This can be done by describing the circles in terms of $(u, v)$ and $\Delta_r$ as follows:

$$\begin{aligned} u^2 + z^2 &= r_1^2 \\ (u - \Delta_r)^2 + z^2 &= r_2^2 \end{aligned} \tag{6.67}$$

The intersecting points must have $u = D_1$; making this substitution and combining the two equations yields:

$$\begin{aligned} (D_1 - \Delta_r)^2 + (r_1^2 - D_1^2) &= r_2^2 \\ D_1 &= \frac{\Delta_r^2 + r_1^2 - r_2^2}{2\Delta_r} \end{aligned} \tag{6.68}$$

$D_2$ can now be solved for by simply substituting $D_1$ into either circle equation and is equal to $\pm\sqrt{r_1^2 - D_1^2}$. In terms of $(u, v)$, the intersection points are thus located at $(D_1, \pm D_2)$. The intersection points can be described in terms of the original coordinate system by applying the same transformation operations that were applied to $(u, v)$ in reverse order; first rotating counter-clockwise by $\theta$, then translating by $(q_1^1, q_1^2)$. This can be expressed as:

$$\mathbf{l}^\star = \begin{bmatrix} q_1^1 \\ q_1^2 \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} D_1 \\ \pm D_2 \end{bmatrix} \tag{6.69}$$

with, $\cos\theta = \frac{q_2^1 - q_1^1}{\Delta_r}$ and $\sin\theta = \frac{q_2^2 - q_1^2}{\Delta_r}$ one has for the final expression:

$$\mathbf{l}^\star = \begin{bmatrix} \frac{1}{2\Delta_r^2}\left\{\left(q_2^1 - q_1^1\right)\left(\Delta_r^2 + r_1^2 - r_2^2\right) \mp \left(q_2^2 - q_1^2\right)\sqrt{4\Delta_r^2 r_1^2 - \left(\Delta_r^2 + r_1^2 - r_2^2\right)^2}\right\} \\ \\ \frac{1}{2\Delta_r^2}\left\{\left(q_2^2 - q_1^2\right)\left(\Delta_r^2 + r_1^2 - r_2^2\right) \pm \left(q_2^1 - s_{1,1}\right)\sqrt{4\Delta_r^2 r_1^2 - \left(\Delta_r^2 + r_1^2 - r_2^2\right)^2}\right\} \end{bmatrix} \tag{6.70}$$

Note this expression is valid only when the circles intersect, or when the following condition is satisfied:

$$\max(r_1, r_2) - \min(r_1, r_2) \leq \Delta_r \leq r_1 + r_2 \tag{6.71}$$

When there is equality for this condition there will be only one solution for $\mathbf{l}^\star$, two if the inequality is strict, and none if it is violated.

Continuing on with this approach, one can in fact develop a closed-form expression for the joint likelihood by simple application of the transformation-of-random-variables approach:

$$f(\mathbf{y}|\mathbf{x}) = \sum_i \frac{f\left(\mathbf{l}^{\star,i}\left[r_1, r_2\right]\right)}{\mid \mathrm{J}(\mathbf{y}) \mid_{\mathbf{l}=\mathbf{l}^{\star,i}[r_1,r_2]}} \tag{6.72}$$

The notation in (6.72) is meant to indicate a sum over all possible solutions of $\mathbf{l}^{\star}[r_1, r_2]$; if no solutions exist then simply $f(\mathbf{y}|\mathbf{x}) = 0$. Additionally, $f\left(\mathbf{l}^{\star,i}\left[r_1, r_2\right]\right)$ here is the joint density of the random variable $\mathbf{l}$ evaluated at the $i$-th solution $\mathbf{l}^{\star,i}[r_1, r_2]$ for given $(r_1, r_2)$ as in (6.66) and as mentioned previously is assumed $\sim \mathcal{N}(\hat{\mathbf{l}}, \mathbf{C})$. The term $\mathrm{J}(\mathbf{y})$ represents the Jacobian matrix of the transformation from $(l_1, l_2)$ to $\mathbf{y}$:

$$\begin{aligned}\mathrm{J}(\mathbf{y}) &= \begin{bmatrix} \frac{\partial y_1}{\partial l_1} & \frac{\partial y_1}{\partial l_2} \\ \frac{\partial y_2}{\partial l_1} & \frac{\partial y_2}{\partial l_2} \end{bmatrix} \\ &= \frac{\left(\Phi^2\alpha^2\left(\|\mathbf{s}_1 - \mathbf{l}\|\|\mathbf{s}_2 - \mathbf{l}\|\right)^{\frac{\alpha}{2}-1}\right)\left(\left(q_1^1 - l_1\right)\left(q_2^2 - l_2\right) - \left(q_1^2 - l_2\right)\left(q_2^1 - l_1\right)\right)}{\left[\|\mathbf{s}_1 - \mathbf{l}\|^{\alpha} + \epsilon\right]^2\left[\|\mathbf{s}_2 - \mathbf{l}\|^{\alpha} + \epsilon\right]^2}\end{aligned} \tag{6.73}$$

While (6.72) can be unwieldy, it is considerably simpler than the marginal likelihood of a single sensor as in (6.13); no special functions or infinite series are required to represent the joint likelihood and it is in fact valid even for non-diagonal estimate covariance matrices $\mathbf{C}$.

A specific scenario is further explored with $\mathbf{w}_1 = [0, 0]^\top$, $\mathbf{q}_2 = [-1, -2]^\top$, $\hat{\mathbf{l}} = [1, 1]^\top$, $\mathbf{C} = \sigma^2\mathbf{I}$ with $\sigma = 0.2$, $\epsilon = 0.5$, $\Phi = 10$, $\alpha = 2$, and where the measurement component from $\mathbf{x}$ is neglected, i.e., $h(\mathbf{q}_1, \mathbf{x}) = h(\mathbf{q}_2, \mathbf{x}) = 0$. A scatter plot of the target samples along with the sensor locations for this scenario is shown in Fig. 6.16. Also plotted in Fig. 6.17 is a scatter plot of the measurement samples alongside the marginal histograms for each component.

Shown in Figs. 6.18 and 6.19 are two views of the expression for the likelihood (6.72) plotted as a solid surface, alongside a 3D histogram of the data (that has been properly scaled) plotted as a wireframe mesh; note there is little discernible difference between the two.

It can be seen from these figures that in this case a strong, albeit nonlinear, correlation exists between separate sensor measurements. Correlation between measurements is actually intuitive, since *if we knew the interferer location*, $\mathbf{l}$, $y_1$ and $y_2$ would in fact be deterministically related via the nonlinear measurement equations. There is however, a complex interplay between the sensor and interferer locations which determines this correlation; while one can generally expect a larger value the closer the two sensors are to one another, it is difficult to predict the influence on correlation the interferer's location has.

Recall this analysis was undertaken without the presence of sensor noise; re-introduction of noise significantly complicates analytical computation of the likelihood function and is not attempted here. However, one can still make the qualitative hypothesis that while the noise will not

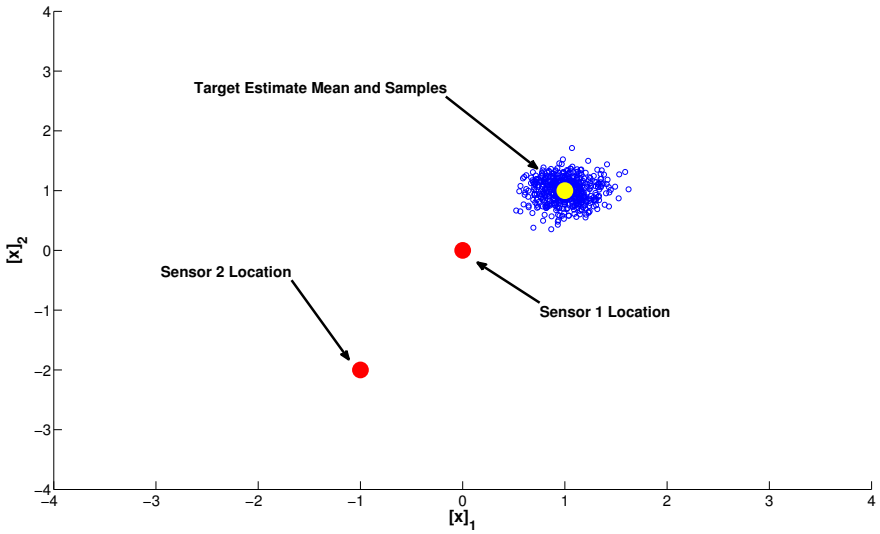


**Figure 6.16:** 2 Sensor 1 Target Scatter Plot for $\mathbf{q}_1 = [0,0]^\top$, $\mathbf{q}_2 = [-1,-2]^\top$, $\bar{\mathbf{l}} = [1,1]^\top$, $\epsilon = 0.5$, $\Phi = 10$, and symmetric estimate covariance $\sigma = 0.2$

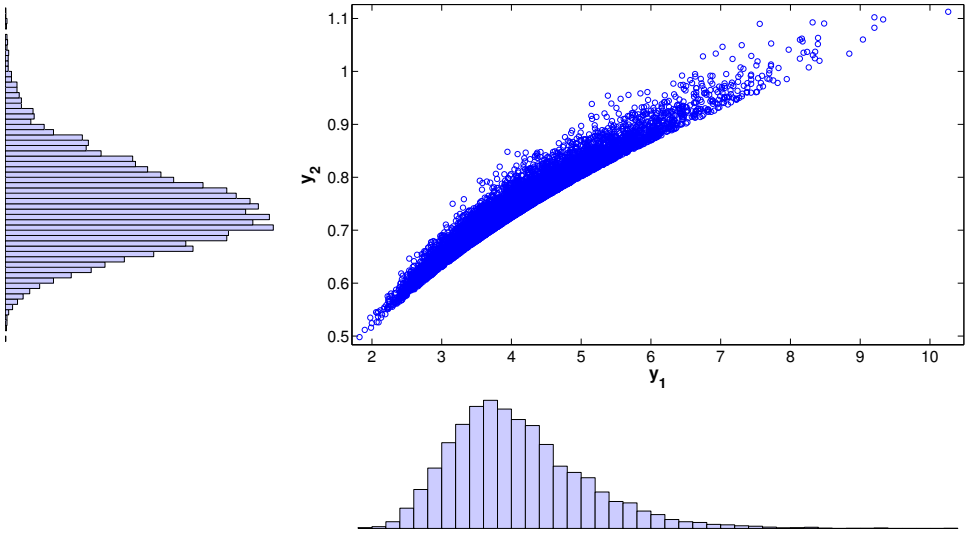


**Figure 6.17:** Measurement Scatter Plot with marginal histograms for 2-Sensor Single Target Scenario

entirely eliminate sensor measurement correlation, it can act to significantly weaken the correlation depending on the relative magnitude of its variance with respect to the measurements. Figure 6.20 shows the original likelihood overlayed in white with a 2D scaled empirical histogram generated using identical parameters as in the previous example, but with independent Gaussian noise ($\sigma = 0.1$) added to both sensor measurements. We immediately see that the sensor noise acts to "blur" the original distribution; the correlation is still discernible but has been significantly diminished.

Developing an exact expression for the measurement likelihood in the most general case, with multiple sensors and multiple foreign targets, represents a daunting task even when drawing upon simplifying assumptions that were made previously. It is nonetheless a desirable prospect, since

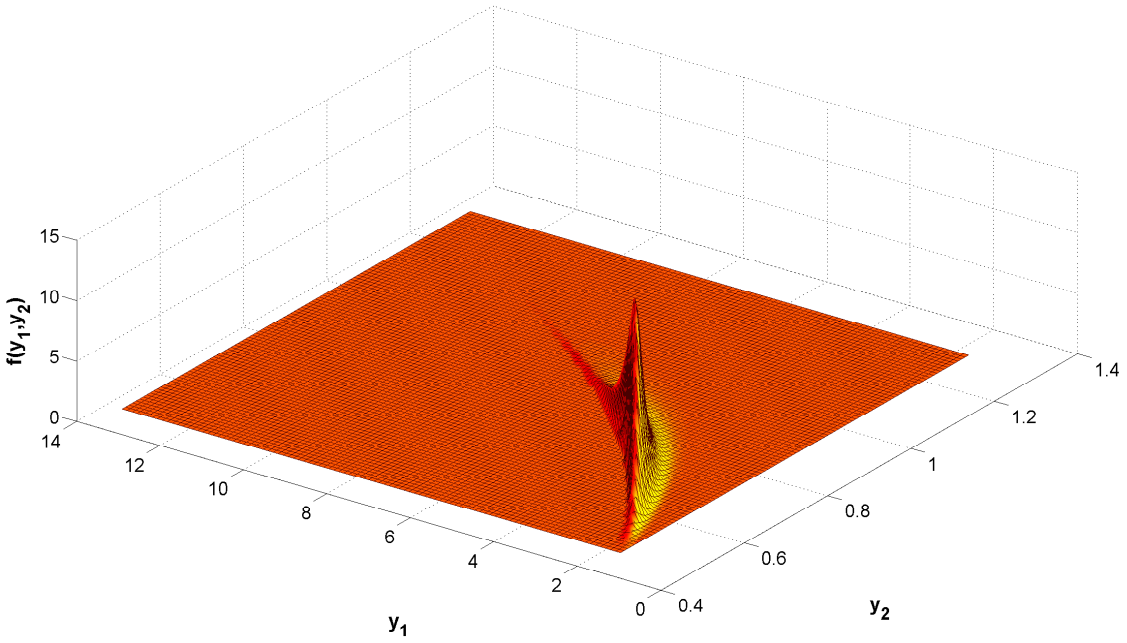


**Figure 6.18:** Distribution Plot for 2-Sensor Single Target Scenario

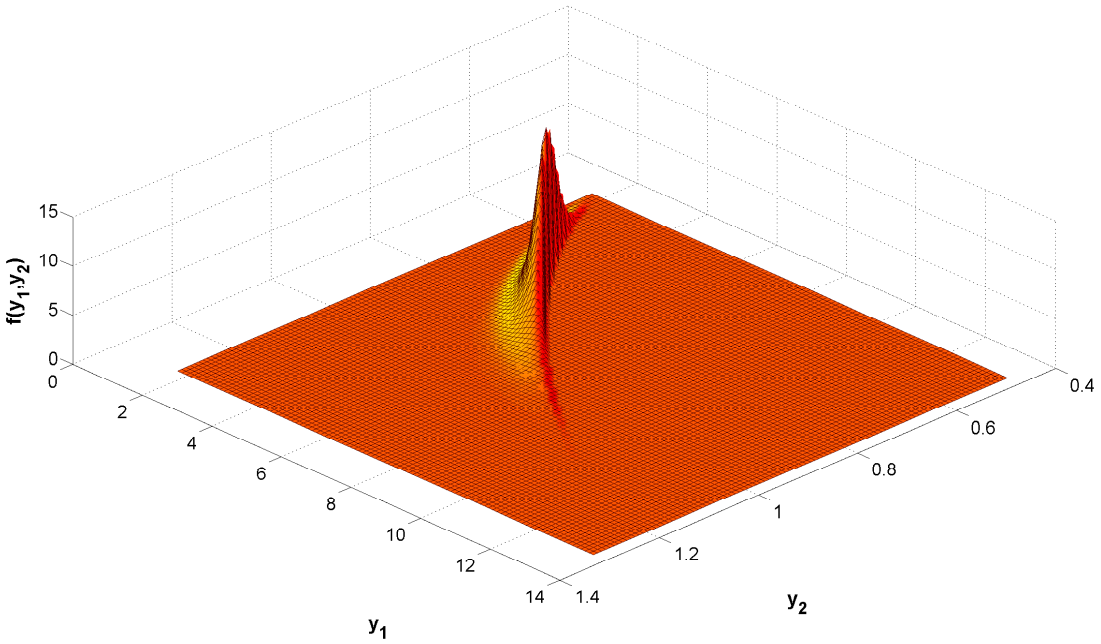


**Figure 6.19:** Second View of Distribution Plot for 2-Sensor Single Target Scenario

possession of such an expression would enable direct computation of the particle weight update equation in (2.37) for the MASCOT algorithm. While it is believed that pursuit of an exact expression with an approach similar to that presented for the noiseless two-sensor single-target

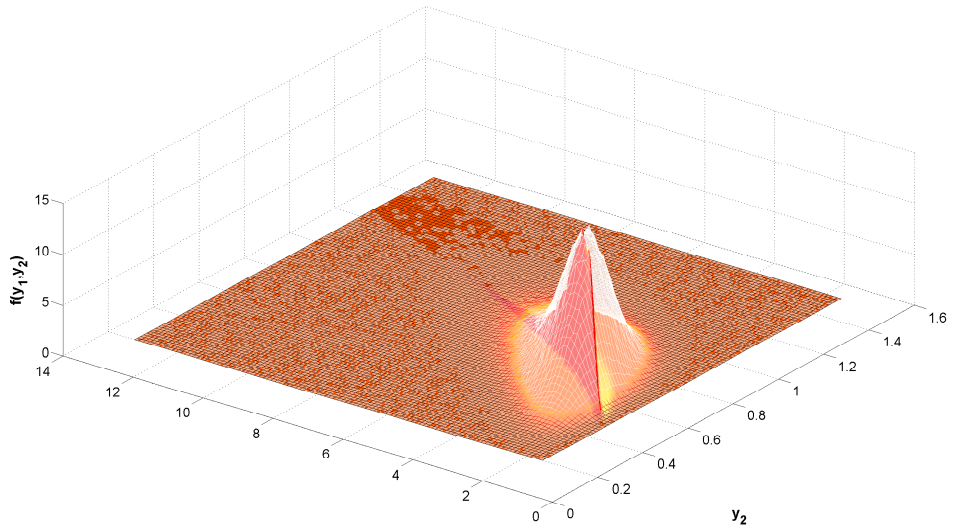


**Figure 6.20:** Two Sensor, Single Target Measurement Distribution Showing the Effects of Sensor Noise

scenario may yield extensions to more complicated scenarios, this is left as an open area for future research.

Arguably, the most important characteristic of the likelihood that should be reproduced as accurately as possible, is the distribution support. This can be seen clearly in the context of a bootstrap particle filter; if the likelihood model does not adequately cover the support of the actual likelihood, then it will be possible for a measurement to occur yielding zero weight for nearly all of the particles, thereby leading to one form of particle degeneracy and subsequent tracking failure. Obviously, it would be ideal for the model to exactly match the truth, but in the case where this is not feasible, it would seem reasonable to proceed conservatively. This is stated in reference to the observation that lacking full knowledge of the likelihood, the model can in some sense be interpreted as a prior distribution. An "informative" prior in this case corresponds to a peaky likelihood with minimal support whereas a flat or conservative prior will spread probability mass across a large region of the likelihood. As was discussed in further detail in a more general context in Chapter 5, there is a tradeoff involved in selection of any prior. Choosing a peakier likelihood can significantly boost tracking performance for outcomes which are more likely to occur, at the elevated risk of track divergence in the case of outliers. To see why this is true, consider the bootstrap particle filter employing a peaky likelihood model and suppose a measurement is taken that corresponds to a specific mean target location. On the average, the filter will strongly favor particles lying close to this location, essentially eliminating particles outside a small neighborhood about the mean. As long as the true target location lies within this neighborhood, and assuming particle degeneracy has not occurred (some particles were indeed propagated into this neighborhood), the estimate variance and error can be very small. The risk manifests in the filter's high sensitivity to deviations from

this ideal situation; measurements that have even a relatively small noise component present can strongly bias the filter away from the target truth causing a large error. There is also an obviously higher risk in particle degeneracy since there is now a much smaller region in the target space that will yield nonzero weights. A flatter likelihood avoids this risk at the expense of reduced "particle differentiation"; a larger region of particles will have nontrivial weights causing a larger variance in the filter estimate (and associated MSE). Thus a balance must be struck between the protection offered by a conservative likelihood which helps mitigate the risk of degeneracy and a "more aggressive" one which can reap larger performance benefits. Nonetheless, one should proceed with caution before introducing any information into the likelihood model. While additional information can be highly beneficial if it is accurate, misrepresentation of the likelihood with an "excessively peaky" model can cause severe performance degradation.

In light of this discussion, let us return to the search for a reasonable representation of the multi-sensor joint likelihood function. As we have seen for the two-sensor, single-target case, the measurements are in general, each dependent on the unknown foreign target locations and thus dependent on each other. It is argued that this dependency represents specific information concerning the joint likelihood, and further argued that one can choose to neglect this information by acting as if the measurements are in fact independent. The independence assumption represents an ignorance about the true form of the likelihood and its underlying inter-dependencies. Doing so allows one to write the likelihood model as:

$$f^{\star}(\mathbf{y}|\mathbf{x}) = \prod_{k=1}^{K} f^{\star}(y_k|\mathbf{x}) \tag{6.74}$$

where this notation is used to make the distinction between the modeled likelihood, $f_{\cdot}^{\star}(\cdot)$, and the true likelihood $f_{\cdot}(\cdot)$. One can loosely assert that $f^{\star}(\mathbf{y}|\mathbf{x}) \approx f_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x})$ under certain conditions, but this is not the prime focus here. The model is meant to serve primarily as a conservative prior, capturing only the information that is indeed possessed about the true likelihood.

The terms $f^{\star}(\mathbf{y}|\mathbf{x})$ represent models of the marginal likelihoods and these can in fact be reasonably approximated using the methods discussed in section 6.0.1. Using equation (6.56) in conjunction with (6.60) and (6.59) we have,

$$\begin{aligned}
f^{\star}(\mathbf{y}|\mathbf{x}) &= \prod_{k=1}^{K} \mathcal{N}\left(y_k \mid \bar{y}_k, \bar{\sigma}_k^2\right) \\
\bar{y}_k &= h\left(\mathbf{q}_k, \mathbf{x}\right) + \sum_{l=1}^{L} \frac{\Phi}{\|\mathbf{d}_{k,l}\|^{\alpha} + \epsilon} \\
\bar{\sigma}_k^2 &= \alpha^2 \Phi^2 \sum_{l=1}^{L} \frac{\|\hat{\mathbf{d}}_{k,l}\|^{2(\alpha-2)} \hat{\mathbf{d}}_{k,l}^{\top} \mathbf{C} \hat{\mathbf{d}}_{k,l}}{\left(\|\hat{\mathbf{d}}_{k,l}\|^{\alpha} + \epsilon\right)^4} + \zeta_k^2 \\
\hat{\mathbf{d}}_{k,l} &= \mathbf{q}_k - \mathbf{l}_l = \begin{bmatrix} q_k^1 - \hat{l}_l^1 & q_k^2 - \hat{l}_l^2 \end{bmatrix}^{\top}
\end{aligned} \tag{6.75}$$

where $\hat{\mathbf{l}}$ is the communicated estimate of the $l$-th foreign target with $\mathbf{C}_l = \begin{bmatrix} \sigma_{l,1}^2 & \rho_l \sigma_{l,1} \sigma_{1,2} \\ \rho_l \sigma_{l,1} \sigma_{1,2} & \sigma_{l,2}^2 \end{bmatrix}$ the corresponding covariance matrix, $\mathbf{q}_k$ is the $k$-th sensor location, and the sensor noise is $\sim \mathcal{N}(0, \zeta_k)$.

Two specific scenarios involving this model are now examined. In both cases, the sensors are positioned at $\mathbf{s}_1 = [0, 0]$ and $\mathbf{s}_2 = [-1, -2]$, with $\epsilon = 0.1$, $\Phi = 10$, $\alpha = 2$, and $\zeta_k = 0.01$. In the first scenario, two interfering targets are located at $\mathbf{l}_1 = [1, 1]$ and $\mathbf{l}_2 = [1, 4]$ with $\mathbf{C}_1 = (0.04)\mathbf{I}$ and $\mathbf{C}_2 = (0.01)\mathbf{I}$. A scatter plot of this scenario is shown in Fig. 6.21. Figure 6.22 shows the scaled 3D empirical histogram as a solid surface, with a mesh overlay representing the Gaussian likelihood model. Although there is little resemblance between the two distributions, the Gaussian approximation does capture a large majority of the histogram's support. This is precisely what was sought to be accomplished; the Gaussian approximation in this case will act as a conservative model for the likelihood.

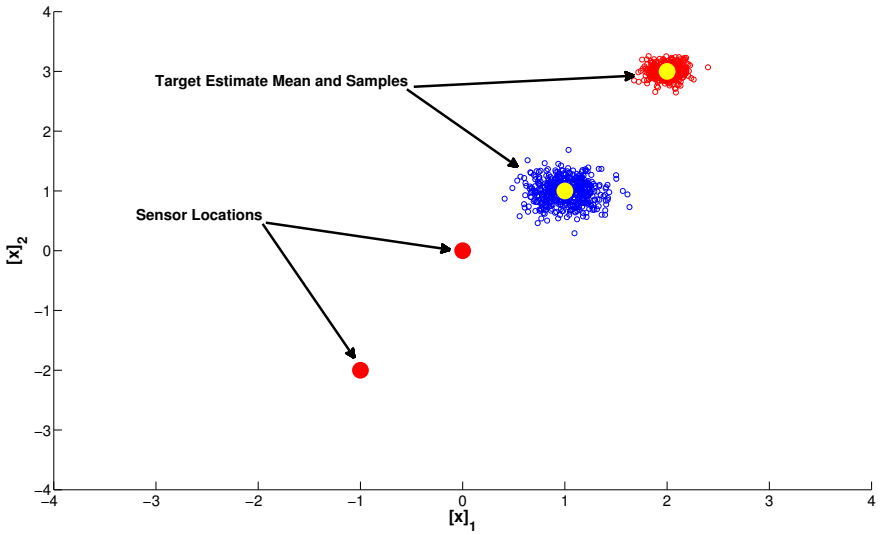


**Figure 6.21:** Two Sensor, Two Target Scatter Plot with $\mathbf{s}_1 = [0, 0]$, $\mathbf{s}_2 = [-1, -2]$,$\epsilon = 0.1$, $\Phi = 10$, $\zeta_k = 0.01$

As the number of foreign targets within the environment increases, it has been found in most cases that the measurement correlation becomes vanishingly small. This is not entirely unexpected; for a given measurement vector $\mathbf{y}$ there are a much larger number of possible target location configurations. The second example considers the case with 6 targets in the configuration as shown in Fig. 6.23 with $\mathbf{C}_1 = \mathbf{C}_4 = (0.04)\mathbf{I}$, $\mathbf{C}_2 = \mathbf{C}_5 = (0.01)\mathbf{I}$, and $\mathbf{C}_3 = \mathbf{C}_6 = (0.09)\mathbf{I}$. The Gaussian model and empirical histogram for this case are plotted in Fig. 6.24, where we can see there is a much closer resemblance between the two distributions than in the two target case; this can be directly attributed to the significantly weakened correlation between measurements resulting from the additional foreign targets.

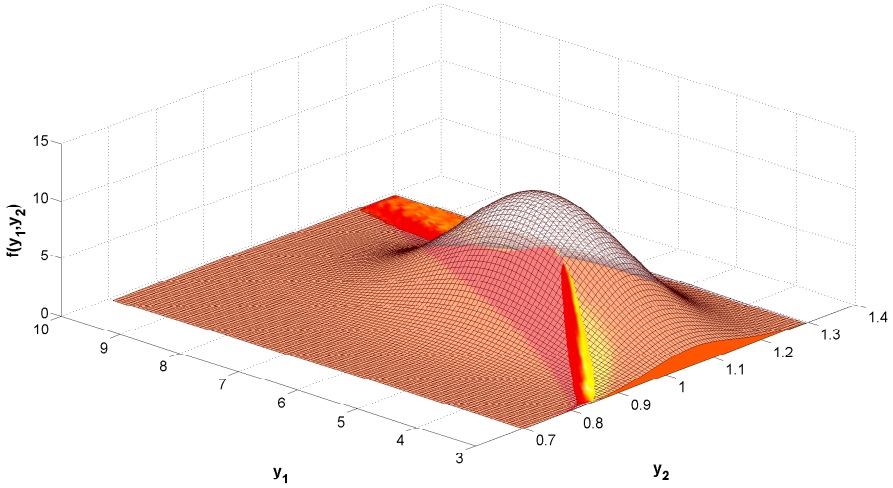


**Figure 6.22:** Two Sensor, Two Target Distribution Plot with $\mathbf{s}_1 = [0, 0]$, $\mathbf{s}_2 = [-1, -2]$,$\epsilon = 0.1$, $\Phi = 10$, $\zeta_k = 0.01$

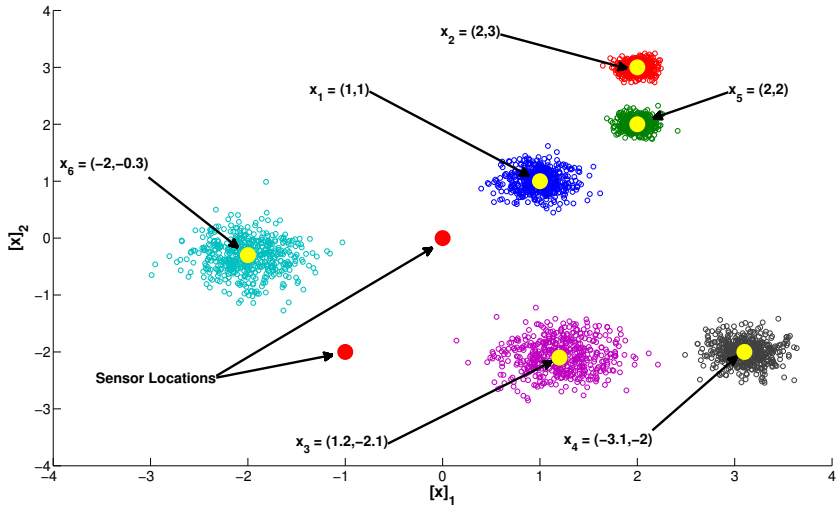


**Figure 6.23:** Two Sensor, Six Target Scatter Plot

It has been shown that the Gaussian model presented in (6.75) offers a reasonable, conservative "first-order" model for the multi-sensor cooperative joint likelihood function under typical tracking conditions. It should also be obvious from the preceding figures that the model can in a sense be interpreted as a multivariate Gaussian fit to the joint likelihood. This interpretation opens up the possibility for model refinements by making adjustments to the Gaussian parameters. It is believed that an approximation can be developed that may be able to capture a "linearized" form of the measurement correlation; this would be incorporated into the model as a non-diagonal covariance matrix (which is Diag$[\bar{\sigma}_k^2]$ in the current model), and thus remains an area open for future possible research.

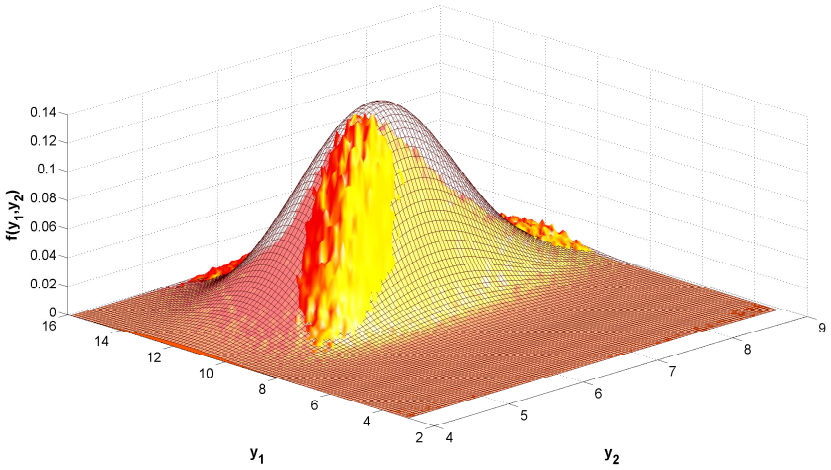


**Figure 6.24:** Two Sensor, Six Target Distribution Plot

# 7

# Conclusion and Future Work

In this dissertation, the MASCOT framework has been introduced and detailed in a generalized setting and further refined for implementation within an MTT environment that involves a distributed network of agents, each endowed with mobile RSSI sensors to cooperatively track the high-dimensional full target space using particle filtering. The framework outlined in 2.2 is complete in the sense that all elements for its implementation have been resolved. Adaptive partitioning of the state space was introduced in that section and further refined for the specific application considered within section 2.6. The problem of agent cooperation was also addressed by first examining the measurement likelihood of a sensor corrupted by interference sources with known Gaussian distributions. A closed form expression was found and subsequent Gaussian approximation developed, allowing its feasible use within an online particle filter. The results of this investigation were leveraged within MASCOT by treating the targets not estimated by a given agent as interferers, with the Gaussian information provided by other agents within the environment. Finally, optimal sensor positioning was then considered for this specific environment. Several solutions were presented for positioning sensors about a single target corrupted by an arbitrary number of interferers. The joint 2-target scenario for sensor positioning was also investigated in detail, admitting a solution similar to the single target case. Performance results demonstrating each separate element of MASCOT were presented and concluded with a scenario demonstrating the final algorithm as outlined in 2.2.

As the focus within this dissertation was on state-estimation, the significant problem of target estimate initialization and detection (which also covers the case whereby the number of targets is dynamic) was not addressed and a known fixed number of targets was assumed. While some work presented here involves a more generalized form of interference compensation which can be leveraged to handle the scenario whereby the number of targets is unknown, this remains a logical and important area for future work. Auxiliary topics parallel to MASCOT regarding tracking using asynchronous measurements and dynamic interference compensation, along with a theoretical investigation of Bayesian performance bounds, were also presented and linked with the main the-

sis. Areas open to future work in these areas were outlined in their respective chapters, with one common objective being their incorporation into the MASCOT framework.

The main system presented is indeed a complete solution for handling a high-dimensional tracking problem and can be extremely effective in mitigating the dimensionality problem while maintaining adequate tracking performance. It is hoped that a future opportunity will arise allowing this system to be implemented in real hardware and applied in an otherwise infeasible environment. It is believed that MASCOT can be extended to any general multi-dimensional estimation problem allowing for a much broader range of application specifically in multi-joint object tracking, financial portfolio management, and medical image analysis.

# References


[1] Petar M. Djurić, Jonathan Beaudeau, and Mónica F. Bugallo. **Non-centralized target tracking with mobile agents**. In *ICASSP*, pages 5928–5931, 2011. 3, 67, 71, 75, 86

[2] M.F. Bugallo, J. Beaudeau, and P.M. Djurić. **A mobile system for non-centralized target tracking in presence of dynamic interferences**. In *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, pages 1–5, Sept 2013. 3

[3] Jonathan Beaudeau, Mónica F. Bugallo, and Petar M. Djurić. **Target tracking with asynchronous measurements by a network of distributed mobile agents**. In *ICASSP*, pages 3857–3860, 2012. 5, 81, 83, 86, 89

[4] Arnaud Doucet and Adam M. Johansen. **A tutorial on particle filtering and smoothing: fifteen years later**, 2011. 8, 16

[5] Arnaud Doucet, Nando Freitas, and Neil Gordon. **An Introduction to Sequential Monte Carlo Methods**. In Arnaud Doucet, Nando Freitas, and Neil Gordon, editors, *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, pages 3–14. Springer New York, 2001. 8

[6] F. Daum and J. Huang. **Curse of dimensionality and particle filters**. In *Aerospace Conference, 2003. Proceedings 2003 IEEE*, **4**, 2003. 8

[7] S. Hug, A. Raue, J. Hasenauer, J. Bachmann, U. Klingmller, J. Timmer, and F.J. Theis. **High-dimensional Bayesian parameter estimation: Case study for a model of JAK2/STAT5 signaling**. *Mathematical Biosciences*, (0):–, 2013. 8

[8] K. Katayama, S. Hagiwara, H. Tsutsui, H. Ochi, and T. Sato. **Sequential importance sampling for low-probability and high-dimensional SRAM yield analysis**. In *Computer-Aided Design (ICCAD), 2010 IEEE/ACM International Conference on*, pages 703–708, 2010. 8

[9] Svante Wold, Kim Esbensen, and Paul Geladi. **Principal component analysis**. *Chemometrics and Intelligent Laboratory Systems*, **2**(13):37 – 52, 1987. ¡ce:title¿Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists¡/ce:title¿. 9

[10] R. E. Kalman. **A New Approach to Linear Filtering and Prediction Problems**. *Transactions of the ASME Journal of Basic Engineering*, (82 (Series D)):35–45, 1960. 14

[11] S. Julier and J. Uhlmann. **A new extension of the Kalman filter to nonlinear systems**. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997. 14

[12] M.F. Bugallo and P.M. Djurić. *Particle Filtering, in Adaptive Signal Processing: Next Generation Solutions (eds T. Adali and S. Haykin)*. John Wiley & Sons Inc, New Jersey, 2010. 16, 72

[13] Michael Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, 1st edition, June 2002. 17

[14] Peter Stone and Manuela Veloso. **Multiagent Systems: A Survey from a Machine Learning Perspective**. *Autonomous Robots*, **8**(3):345–383, June 2000. 17

[15] V. Gorodetski and I. Kotenko. **The multi-agent systems for computer network security assurance: frameworks and case studies**. In *Artificial Intelligence Systems, 2002. (ICAIS 2002). 2002 IEEE International Conference on*, pages 297–302, 2002. 17

[16] Rosa M Vicari, Cecilia D Flores, Andr M Silvestre, Louise J Seixas, Marcelo Ladeira, and Helder Coelho. **A multi-agent intelligent environment for medical knowledge**. *Artificial Intelligence in Medicine*, **27**(3):335 – 366, 2003. ¡ce:title¿Software Agents in Health Care¡/ce:title¿. 17

[17] Victor Lesser, Milind Tambe, and Charles L. Ortiz, editors. *Distributed Sensor Networks: A Multiagent Perspective*. Kluwer Academic Publishers, Norwell, MA, USA, 2003. 17

[18] Juan Liu, M. Chu, Jie Liu, J. Reich, and Feng Zhao. **Distributed state representation for tracking problems in sensor networks**. In *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*, pages 234–242, 2004. 18, 19

[19] Juan Liu, M. Chu, and J.E. Reich. **Multitarget Tracking in Distributed Sensor Networks**. *Signal Processing Magazine, IEEE*, **24**(3):36–46, 2007. 18

[20] G.W. Pulford. **Taxonomy of multiple target tracking methods**. *Radar, Sonar and Navigation, IEE Proceedings -*, **152**(5):291–304, 2005. 19

[21] Ronald P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Inc., Norwood, MA, USA, 2007. 19

[22] Ronald P Mahler. **A Theoretical Foundation for the Stein-Winter" Probability Hypothesis Density (PHD)" Multitarget Tracking Approach**. Technical report, DTIC Document, 2000. 19

[23] M.F. Bugallo, Ting Lu, and P.M. Djurić. **Target Tracking by Multiple Particle Filtering**. In *Aerospace Conference, 2007 IEEE*, pages 1–7, 2007. 19, 22, 23

[24] M.F. Bugallo and P.M. Djurić. **Target tracking by symbiotic particle filtering**. In *Aerospace Conference, 2010 IEEE*, pages 1–7, 2010. 19, 59

[25] O. Hlinka, F. Hlawatsch, and P.M. Djurić. **Distributed particle filtering in agent networks: A survey, classification, and comparison**. *Signal Processing Magazine, IEEE*, **30**(1):61–81, 2013. 23

[26] M.F. Bugallo and P.M. Djuric. **Particle filtering in high-dimensional systems with Gaussian approximations**. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 8013–8017, May 2014. 29

[27] Chris Snyder, Thomas Bengtsson, Peter Bickel, and Jeff Anderson. **OBSTACLES TO HIGH-DIMENSIONAL PARTICLE FILTERING**. 29

[28] F. Daum and J. Huang. **Curse of dimensionality and particle filters**. In *Aerospace Conference, 2003. Proceedings. 2003 IEEE*, **4**, pages 1979–1993, March 2003. 29

[29] P. Bui Quang, C. Musso, and F. Le Gland. **An insight into the issue of dimensionality in particle filtering**. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8, July 2010. 29

[30] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. Sukhatme. **Robomote: enabling mobility in sensor networks**. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 404–409, 2005. 35

[31] S. Martinez and F. Bullo. **Optimal sensor placement and motion coordination for target tracking**. *Automatica*, **42**(4):661–668, April 2006. 37

[32] R. Olfati-Saber. **Distributed Tracking for Mobile Sensor Networks with Information-Driven Mobility**. In *American Control Conference, 2007. ACC '07*, pages 4606–4612, 2007. 37

[33] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte. **Information-theoretic coordinated control of multiple sensor platforms**. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, **1**, pages 1521–1526 vol.1, 2003. 37

[34] Hu Haifeng and Yang Zhen. **Mobile-Agent-Based Information-Driven Multiresolution Algorithm for target tracking in Wireless Sensor Networks**. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on*, **1**, pages 521–525, 2007. 37

[35] Yi Zou and K Chakrabarty. **Distributed Mobility Management for Target Tracking in Mobile Sensor Networks**. *Mobile Computing, IEEE Transactions on*, **6**(8):872–887, 2007. 37

[36] S. Maheswararajah and S. Halgamuge. **Mobile Sensor Management For Target Tracking**. In *Wireless Pervasive Computing, 2007. ISWPC '07. 2nd International Symposium on*, pages –, 2007. 37

[37] David Moreno-Salinas, Antonio M. Pascoal, and Joaquin Aranda. **Optimal Sensor Placement for Multiple Target Positioning with Range-Only Measurements in Two-Dimensional Scenarios**. *Sensors*, **13**(8):10674–10710, 2013. 37, 39, 47

[38] Yao Li and P.M. Djurić. **Particle Filtering for Target Tracking with Mobile Sensors**. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, **2**, pages II–1101–II–1104, 2007. 37

[39] G. Boger. *Multiplicative Programming: Theory and Algorithms*. University of Florida, 1999. 40

[40] Nguyen Van Thoai. **A global optimization approach for solving the convex multiplicative programming problem**. *Journal of Global Optimization*, **1**(4):341–357, 1991. 40

[41] Hanif D. Sherali and Hongjie Wang. **Global optimization of nonconvex factorable programming problems**. *Mathematical Programming*, **89**(3):459–478, 2001. 40

[42] Abdullah Konak, David W Coit, and Alice E Smith. **Multi-objective optimization using genetic algorithms: A tutorial**. *Reliability Engineering & System Safety*, **91**(9):992–1007, 2006. 40

[43] Konstantinos E Parsopoulos and Michael N Vrahatis. **Particle swarm optimization method in multiobjective problems**. In *Proceedings of the 2002 ACM symposium on Applied computing*, pages 603–607. ACM, 2002. 40

[44] Louis L. Scharf and L. T. McWhorter. **Geometry of the Cramer-Rao Bound**. *Signal Process.*, **31**(3):301–311, April 1993. 48

[45] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*. Prentice-Hall Inc, New Jersey, 1993. 71

[46] h B.A. Yocom, B.R. La Cour, and T.W. Yudichak. **A Bayesian Approach to Passive Sonar Detection and Tracking in the Presence of Interferers**. *Oceanic Engineering, IEEE Journal of*, **36**(3):386–405, 2011. 73

[47] B. Friedland. **Treatment of bias in recursive filtering**. *Automatic Control, IEEE Transactions on*, **14**(4):359–367, 1969. 73

[48] B. Friedland. **Recursive filtering in the presence of biases with irreducible uncertanty**. *Automatic Control, IEEE Transactions on*, **21**(5):789–790, 1976. 73

[49] Xiangdon Lin and Y. Bar-Shalom. **Multisensor target tracking performance with bias compensation**. *Aerospace and Electronic Systems, IEEE Transactions on*, **42**(3):1139–1149, 2006. 73

[50] Y.Y. Hu and D.H. Zhou. **Bias fusion estimation for multi-target tracking systems with multiple asynchronous sensors**. *Aerospace Science and Technology*, **27**(1):95 – 104, 2013. 73

[51] "Yu GAO, Jian qiu ZHANG, and Bo HU". **A Polynomial Prediction Filter Method for Estimating Multisensor Dynamically Varying Biases**. *Chinese Journal of Aeronautics*, **20**(3):240 – 246, 2007. 73

[52] Mónica F Bugallo, Ting Lu, and Petar M Djurić. **FUSION OF INFORMATION FROM BIASED SENSOR DATA BY PARTICLE FILTERING**. 2007. 73

[53] Mónica F Bugallo, Ting Lu, and Petar M Djurić. **Tracking with biased measurements of signal strength sensors**. In *Digital Signal Processing, 2007 15th International Conference on*, pages 567–570. IEEE, 2007. 73

[54] MF Bugallo, Ting Lu, and PM Djurić. **Bearings-only tracking with biased measurements**. In *Computational Advances in Multi-Sensor Adaptive Processing, 2007. CAMPSAP 2007. 2nd IEEE International Workshop on*, pages 265–268. IEEE, 2007. 73

[55] GEORGE CASELLA and CHRISTIAN P. ROBERT. **Rao-Blackwellisation of sampling schemes**. *Biometrika*, **83**(1):81–94, 1996. 74

[56] F. Mustiere, Miodrag Bolic, and M. Bouchard. **A Modified Rao-Blackwellised Particle Filter**. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, **3**, pages III–III, 2006. 74

[57] M.F. Bugallo, J. Beaudeau, and P.M. Djurić. **A mobile system for non-centralized target tracking in presence of dynamic interferences**. In *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, pages 1–5, Sept 2013. 74

[58] Y. Bar-Shalom. **Update with out-of-sequence measurements in tracking: exact solution**. *Aerospace and Electronic Systems, IEEE Transactions on*, **38**(3):769–777, 2002. 81

[59] M. Mallick, S. Coraluppi, and C. Carthel. **Advances in asynchronous and decentralized estimation**. In *Aerospace Conference, 2001, IEEE Proceedings.*, **4**, pages 4/1873–4/1888 vol.4, 2001. 81

[60] Ruixin Niu, P. Varshney, K. Mehrotra, and C. Mohan. **Temporal fusion in multi-sensor target tracking systems**. In *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, **2**, pages 1030–1037 vol.2, 2002. 81

[61] Dimitris K. Tasoulis, Niall M. Adams, and David J. Hand. **Selective fusion of out-of-sequence measurements**. *Information Fusion*, **11**(2):183 – 191, 2010. 81

[62] Guo Wenyan and Han Chongzhao. **Interpolation particle filter for tracking with out-of-sequence measurements**. In *Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on*, pages 2254–2258, 2008. 81

[63] B.N. Oreshkin and M.J. Coates. **Asynchronous distributed particle filter via decentralized evaluation of Gaussian products**. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8, 2010. 81

[64] ngel F. Garca-Fernndez and Jess Grajal. **Asynchronous particle filter for tracking using non-synchronous sensor networks**. *Signal Processing*, **91**(10):2304 – 2313, 2011. 81, 85

[65] C. R. Rao. **Information and accuracy attainable in the estimation of statistical parameters**. *Bull.Cal.Math.Soc.*, **37**:81–91, 1945. 91

[66] H. L. Van Tree. *Detection, Estimation and Modulation Theory*. John Wiley and Sons Inc, New York, 1968. 91

[67] Harry L. Van Trees and Kristine L. Bell. *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*. Wiley-IEEE Press, 2007. 91, 110

[68] T.Y. Young and R. Westerberg. **Error bounds for stochastic estimation of signal parameters**. *Information Theory, IEEE Transactions on*, **17**(5):549–557, 1971. 93, 94

[69] Z. Ben-Haim and Y.C. Eldar. **A Lower Bound on the Bayesian MSE Based on the Optimal Bias Function**. *Information Theory, IEEE Transactions on*, **55**(11):5179–5196, 2009. 93, 94

[70] I. M. Gelfand and S. V. Fomin. *Calculus of Variations (Dover Books on Mathematics)*. Dover Publications, October 2000. 94

[71] Hidekazu Tanaka and Masafumi Akahira. **On a family of distributions attaining the Bhattacharyya bound**. *Annals of the Institute of Statistical Mathematics*, **55**(2):309–317, 2003. 96, 110

[72] K. Todros and J. Tabrikian. **Achievable MSE lower bounds in non-Bayesian Biased estimation**. In *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2010 IEEE*, pages 117–120, 2010. 96, 110

[73] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory (v. 1)*. Prentice Hall, 1 edition, April 1993. 98

[74] A.O. Hero, J.A. Fessler, and M. Usman. **Exploring estimator bias-variance tradeoffs using the uniform CR bound**. *Signal Processing, IEEE Transactions on*, **44**(8):2026–2041, 1996. 98

[75] S. Kay and Y.C. Eldar. **Rethinking biased estimation [Lecture Notes]**. *Signal Processing Magazine, IEEE*, **25**(3):133–136, 2008. 98

[76] Yufei Huang and J. Zhang. **Lower bounds on the variance of deterministic signal parameter estimators using Bayesian inference**. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, **6**, pages VI–745–8 vol.6, 2003. 99, 100, 101, 112

[77] A. Host-Madsen. **On the existence of efficient estimators**. *Signal Processing, IEEE Transactions on*, **48**(11):3028–3031, 2000. 109

[78] P. Tichavsky, C.H. Muravchik, and Arye Nehorai. **Posterior Cramer-Rao bounds for discrete-time nonlinear filtering**. *Signal Processing, IEEE Transactions on*, **46**(5):1386–1396, 1998. 110, 114, 115

[79] N. Bergman and Universitetet i Linköping. Dept. of Electrical Engineering. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Department of Electrical Engineering, Linköping University, 1999. 110

[80] Miroslav imandl, Jakub Krlovec, and Petr Tichavsk. **Filtering, predictive, and smoothing CramrRao bounds for discrete-time nonlinear dynamic systems**. *Automatica*, **37**(11):1703 – 1716, 2001. 110

[81] J.H. Taylor. **The Cramer-Rao estimation error lower bound computation for deterministic nonlinear systems**. *Automatic Control, IEEE Transactions on*, **24**(2):343–344, 1979. 110

[82] Sudha Challa. *Fundamentals of object tracking / Subhash Challa... [et al.]*. Cambridge University Press Cambridge ; New York, 2011. 115

[83] F. Gustafsson and A.J. Isaksson. **Best choice of coordinate system for tracking coordinated turns**. In *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, **3**, pages 3145–3150 vol.3, 1996. 123

[84] D. C. Gilliland. **Integral of the Bivariate Normal Distribution over an Offset Circle**. *Journal of the American Statistical Association*, **57**(300):758–768, 1962. 131

[85] D. C. Gilliland and E. R. Hansen. **A note on some series representations of the integral of a bivariate normal distribution over an offset circle**. *Naval Research Logistics Quarterly*, **21**(1):207–211, 1974. 131

[86] S. Kotz, N. L. Johnson, D. W. Boyd, and TORONTO UNIV (Ont.) DEPT OF INDUSTRIAL ENGINEERING. *Series Representations of Distributions of Quadratic Forms in Normal Variables, Ii. Noncentral Case*. Defense Technical Information Center, 1966. 131

[87] H. Weil. **The Distribution of Radial Error**. *Annals of Mathematical Statistics*, **25**(1):168–170, 1954. 131

[88] V. Massidda. **Analytical Calculation of a Class of Integrals Containing Exponential and Trigonometric Functions**. *Mathematics of Computation*, **41**(164):pp. 555–557, 1983. 131

[89] W Fulks. **A generalization of Laplace's method**. *Proceedings of the American Mathematical Society*, **2**(4):613–622, 1951. 146